ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
«ҒҰМАРБЕК ДӘУКЕЕВ АТЫНДАҒЫ АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ»
коммерциялық емес акционерлік қоғамы
Ақпарттық қауіпсіздік жүйелері кафедрасы

«ҚОРҒАУҒА ЖІБЕРІЛДІ»
Кафедра меңгерушісі с.ғ.к., доцент Бердібаев Р.Ш.
(ғылыми дәрежесі, атағы, аты-жөні)
_____ «_____» _____ 2020 ж.
(қолы)

ДИПЛОМДЫҚ ЖОБА

Тақырыбы: MD5 шифрлау әдісін қолдану арқылы қауіпсіз веб-қосымшаны әзірлеу

Мамандығы: 5В100200 – «Ақпараттық қауіпсіздік жүйелері»
Орындаған: Алдашов Нуржан Талгатович Тобы: СИБк-16-1
(аты-жөні)
Ғылыми жетекші: т.ғ.к., доцент Шайкулова Актоты Алиевна
(ғылыми дәрежесі, атағы, аты-жөні)
Кеңесшілер:
Мамандығы бойынша:
аға оқытушы Дмитриева Маргарита Валерьевна
(ғылыми дәрежесі, атағы, аты-жөні)
_____«_____»_____ 2020 ж.
(қолы)
Тіршілік қауіпсіздігі бөлімі бойынша:
доцент Жандаулетова Фарида Рустембековна
(ғылыми дәрежесі, атағы, аты-жөні)
_____«_____»_____ 2020 ж.
(қолы)
Мөлшер бақылаушы:
аға оқытушы Альмуратова Камшат Бимуратовна
(ғылыми дәрежесі, атағы, аты-жөні)
_____«_____»_____ 2020 ж.
(қолы)
Пікір беруші:
Төлеулиев Сырым Бимуратович
(ғылыми дәрежесі, атағы, аты-жөні)
_____«_____»_____ 2020 ж.
(қолы)

Алматы 2020

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
«ҒҰМАРБЕК ДӘУКЕЕВ АТЫНДАҒЫ АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ»
коммерциялық емес акционерлік қоғамы

Басқару және ақпараттық технологиялар институты
Ақпараттық қауіпсіздік жүйелері кафедрасы
5В100200 – «Ақпараттық қауіпсіздік жүйелері» мамандығы

Дипломдық жобаны орындауға берілген
ТАПСЫРМА

Студент: Алдашов Нуржан Талгатович
(аты-жөні)

Жобаның тақырыбы: MD5 шифрлау әдісін қолдану арқылы қауіпсіз веб-қосымшаны әзірлеу

2019 ж. «11» қараша №146 университет бұйрығымен бекітілді.

Аяқталған жұмысты тапсыру мерзімі: « 5 » маусым 2020 ж.

Жобаға алғашқы деректер (талап етілетін зерттеу (жоба) нәтижелерінің параметрлері және зерттеу нысанының алғашқы деректері):
Веб-қосымша арқылы қолданушы енгізген ақпаратты деректер қорында сенімді сақтау үдерісін жүзеге асыру үшін MD5 шифрлау алгоритмі қолданылды. Веб-қосымшаның клиенттік және серверлік әзірлеу технологиялары: HTML, CSS, JavaScript, PHP, OpenServer пайданылды.

Диплом жобасындағы әзірленуі тиіс мәселелер тізімі немесе диплом жобасының қысқаша мазмұны:
1. Веб-қосымшаның құру кезеңдері.
2. Клиенттік және серверлік әзірлеу технологиялары.
3. Веб-қосымшалардың қауіпсіздік шаралары.
4. MD5 шифрмен парольдік қорғау.
5. Ақпараттық қауіпсіздік тәуекелдерін екі параметр бойынша бағалау.
6. Жұмыс жағдайында желдету жүйесі және өрт қауіпсіздігін есептеу.

Графикалық материалдардың (міндетті түрде дайындалатын сызбаларды көрсету) тізімі:

1.9 сурет – Жергілікті серверді деректер қорына қосу үдерісі

2.12 сурет – MD5 алгоритмінің блок сұлбасы

3.1 сурет – Әрекеттесу алгоритмінің бастапқы қадамы

3.2 сурет – Әрекеттесу алгоритмінің аяқталуы

3.5 сурет – Басты бөлімнің құрылымы

3.17 сурет – Төлемнің деректер қорына түсуі

3.18 сурет – Қолданушы енгізген ақпараттарды тексеру

4.2 сурет – MD5 шифрмен парольдік қорғау

4.2 - кесте – Тәуекелдерді бағалаудың қорытынды кестесі

5.2 - кесте – Кондиционердің негізгі техникалық сипаттамалары

Негізгі ұсынылатын әдебиеттер:

1. Джоел С., Майк Ш. Секреты хакеров. Безопасность Web-приложений – готовые решения. :Пер.с англ. – М.: Издательский дом “Вильямс”,2003. – 384 с.

2. Книга веб-программиста. Секреты профессиональной разработки веб-сайтов / Б.Хоган и др. - Москва: Мир, 2013. - 288 с.

3. Абикенова А.А., Санатова Т.С. Безопасность жизнедеятельности. Методические указания к выполнению раздела «Пожарная профилактика» в выпускных работах для всех специальностей. Бакалавриат - Алматы: АИЭС, 2009. - 32 с.

Жоба бойынша жобаның бөлімдеріне қатысты белгіленген кеңесшілер

Бөлімдері	Кеңесшілері	Мерзімі	Қолы
Негізгі бөлім	Шайкулова А.А.	03.03.2020ж	
А.Қ.Т.Е.	Дмитриева М.В.	13.04.2020ж	
Ө.Т.Қ.Н.	Жандаулетова Ф.Р.	20.04.2020ж	
Нормабақылаушы	Альмуратова К.Б.	02.06.2020ж	

Диплом жобасын дайындау
КЕСТЕСІ

Бөлімдердің атауы, әзірленетін мәселелердің тізімі	Ғылыми жетекшіге ұсыну мерзімдері	Ескерту
Кіріспе	12.02.20 – 15.02.20	орындалды
1.1 Веб-қосымшаның құру кезеңдері	18.02.20 – 25.02.20	орындалды
1.2 Клиенттік және серверлік әзірлеу технологиялары	26.02.20 – 10.03.20	орындалды
2 Веб-қосымшалардың қауіпсіздік шаралары	13.03.20 – 21.03.20	орындалды
3 MD5 шифрмен парольдік қорғау	27.03.20 – 18.04.20	орындалды
4 Ақпараттық қауіпсіздіктің тәуекелдерін есептеу	25.04.20 – 12.05.20	орындалды
4.1 Ақпараттық қауіпсіздік тәуекелдері	19.04.20 – 01.05.20	орындалды
4.2 Екі параметр бойынша есептеу	03.05.20 – 14.05.20	орындалды
5 Өміртіршілік қауіпсіздігі	08.05.20 – 28.05.20	орындалды
5.1 Кәсіпорындағы еңбек жағдайларын талдау	08.05.20 – 15.05.20	орындалды
5.2 Есептеу бөлімі	15.05.20 – 28.05.20	орындалды

Тапсырманың берілген уақыты «12» қаңтар 2020 ж.

Кафедра меңгерушісі _____ (Бердібаев Р.Ш.)
(қолы) (аты-жөні)

Жобаның
ғылыми жетекшісі _____ (Шайкулова А. А.)
(қолы) (аты-жөні)

Орындалатын тапсырманы
қабылдаған студент _____ (Алдашов Н.Т.)
(қолы) (аты-жөні)

Аңдатпа

Бұл дипломдық жобада MD5 шифрлау әдісін қолдану арқылы қауіпсіз веб-қосымша әзірленді. Веб-қосымшаның клиенттік бөлігін әзірлеу барысында қауіпсіздікті қамтамасыз ету үшін HTML, CSS, JavaScript әзірлеу технологиялары қолданылды, ал серверлік жағында PHP бағдарламалық тілі және MySQL деректер қоры қолданылды. Сондай-ақ веб-қосымшаның қауіпсіздігін тексеру үшін OpenServer жергілікті сервер пайдаланылды.

Дипломдық жобаның негізгі мақсаты веб-қосымшаның қауіпсіздігін әзірлеу. Веб-қосымшаның әзірлеу құралдары, арнайы бағдарламалау тілдері, этаптары мен технологиялары қарастырылды. Толық түрде веб-қосымшалардың қорғау түрлері сипатталды, қорғау түрлерінің кемшіліктері және тиімді жақтары анықталды. Нәтижесінде веб-қосымшаны әзірлеу барысында ақпаратты MD5 шифрлау әдісі арқылы корғау үдерісі іске асырылады.

Аннотация

В данном дипломном проекте было разработано безопасное веб-приложение с использованием метода шифрования MD5. Для обеспечения безопасности при разработке клиентской части веб-приложения использовались технологии разработки HTML, CSS, JavaScript, а на серверной стороне использовались программный язык PHP и база данных MySQL. Также для проверки безопасности веб-приложения был использован локальный сервер OpenServer.

Основной целью дипломного проекта является разработка безопасности веб-приложения. Были рассмотрены инструменты разработки веб-приложений, специальные языки программирования, этапы и технологии. Подробно были описаны виды защиты веб-приложений, выявлены недостатки и эффективные стороны форм защиты. В результате при разработке веб-приложения осуществляется процесс защиты информации методом шифрования MD5.

Annotation

In this diploma project, a secure web application was developed using the MD5 encryption method. To ensure security, the client side of the web application was developed using HTML, CSS, and JavaScript development technologies, while the server side used the PHP programming language and the MySQL database. The local OpenServer server was also used to check the security of the web application.

The main goal of the diploma project is to develop the security of a web application. Web application development tools, special programming languages, stages and technologies were considered. The types of protection for web applications were described in detail, and the disadvantages and effective aspects of the forms of protection were identified. As a result, when developing a web application, information is protected using MD5 encryption.

Мазмұны

Кіріспе

Қазіргі заманда ғаламтордың күнделікті даму қарқынына байланысты веб-қосымшалардың желі ішінде таралу жылдамдығына әсер етеді. Көптеген компаниялар веб-қосымша арқылы өзіндік өнімдерді сатуға және компания ішіндегі бизнес үдерістерді толық басқаруға мүмкіндік береді. Әдетте өндірістік немесе коммерциялық компаниялар веб-қосымшаны ішкі және сыртқы мақсатта жиі пайдаланады. Веб-қосымша ішкі мақсатта өнімнің тиімді сатылуында, оңай түрде ақпаратты өңдеп, компания өнімінің сатылу пайызын көбейтуге қолданады. Ал сыртқы мақсатта компания веб-қосымшаларды қолданушылардың саның арттыруға және өнімнің ғаламтор желі ішінде танымалдылығына негізделеді.

Бүгінгі күні ақпараттық жүйелердің веб-қосымшаларға ұқсас түрде құрылуына байланысты, қауіпсіздік мәселесіне аса көп көңіл бөлінуде. Веб-қосымшалардың ең маңызды міндеттері ретінде ақша ресурстарымен алмасу, сақтау, өңдеу қызметтері атқарылады [1]. Осыған байланысты қолданушы веб-қосымшамен әрекеттесу кезеңінен бастап серверден жауап қайтару үдерісі арасындағы қауіпсіздікті қамтамасыз ету қажет.

Ғаламтордың тұрақты түрде және жылдам жұмыс істеуіне байланысты көптеген бизнестің маңызды элементтерінің веб-қосымшаларға ауысуы орын алуда. Әртүрлі веб элементтердің ғаламторда кең таралуына байланысты зиянкестер үшін тартымды нысан болып отыр, сондықтан веб-ресурстарды тиімді қорғау жөніндегі шешімдер қазіргі уақытта өзекті және сұранысқа ие болып табылады.

Осы жобаның өзектілігі, қолданушы енгізген жеке ақпаратты MD5 шифрлау алгоритмі арқылы сенімді қорғаныс түрін қамтамасыздандырып, веб-қосымшаны әзірлеу технологияларын дұрыс пайдалану мәселесі болып табылады.

Дипломдық жобаның негізгі мақсаты – веб-қосымшаның қауіпсіздігін MD5 шифрлау әдісі арқылы қамтамасыз ету. Жоба бойынша веб-қосымшаның әзірлеу құралдары, арнайы бағдарламалау тілдері, кезеңдері мен технологиялары қарастырылды. Толық түрде веб-қосымшалардың қорғау түрлері сипатталды, қорғау түрлерінің кемшіліктері және тиімді жақтары анықталды. HTML, CSS, JavaScript және PHP технологиялары қолданылып, деректер қорымен байланыс орнатылды. Нәтижесінде құпия ақпаратты MD5 шифрлау әдісі арқылы корғау үдерісі іске асырылады.

1 Тақырыптық салаға шолу

Жұмыстың негізгі мақсаты Веб-қосымшаның қауіпсіздігін әзірлеу. Оған жататындар:

1) веб-қосымшаның құру құралдарын, арнайы бағдарламалау тілдері, кезеңдері мен технологияларын қарастыру;

2) веб-қосымшалардың қорғау түрлерін талдап, талдау барысында кемшіліктері және тиімді жақтарын анықтау;

3) қосымшалардың осал тұстарына жасалатын шабуылдардың түрлерін талдау;

4) қауіпті шабуылдар түрлерін талдау арқылы, веб-қосымшалардың қауіпсіздігін арттыру және шабуыл қауіпін азайту;

5) веб-қосымшаны әзірлеу барысында ақпаратты шифрлау арқылы корғау үдерісі іске асырылады.

1.1 Веб-қосымша түсінігі

Веб-қосымша ұғымы – дегеніміз клиент пен сервердің өзара әрекеттесу арқылы жүзеге асырылатын үдеріс. Қолданушы веб-қосымша арқылы сұраныс жіберу арқылы веб-серверден қажет ақпаратты алады.

Клиент және серверлік қосымшаның негізгі бөлігі қашықтағы серверде орналасады, ал қолданушының интерфейсі шолғыш арқылы веб-парақшалар ретінде пайдаланушыларға ұсынылады. Веб-қосымшаны іске қосу үшін қолданушы арнайы бағдарламаларды орнатуды қажет етпейді, ол кез-келген құрылғының шолғышы арқылы ғаламторға қосыла алады. Клиенттің жұмысы пайдаланушының құрылғысындағы операциялық жүйеге байланысты болмайды, сондықтан веб-қосымшаларды әзірлеу операциялық жүйелер үшін арнайы нұсқаларды жазудың қажет етпейді.

Веб-қосымшаның бастауы World Wide Web жобадан ғаламтор беткейлеріне таралған болатын. Бұл жоба бастапқыда серверлер арасында таратылған ақпараттарды іздеуді жеңілдетуге арналған құрал ретінде ойластырылған. World Wide Web ғаламтордың ажырамас бөлшегі болғандықтан, веб-шолғыштар деп аталатын бағдарламалық топтар құрастырылған. Веб-қосымшалар ғаламторда толық кедергілерсіз таралу үшін веб-протоколдар және жәлілік форматтарды құрып, қолдануды үйлестіру үшін World Wide Web Consortium технологиясы қарастырылды. Бұл технология веб-қосымшалардың келесі бірқалыпты стандарттарға жүгіну мүмкіндіктерін қалыптастырды:

– веб-қызметтерді дамыту және кез келген жерден осы деректерге қол жетімділікті қамтамасыз ету;

– ақпараттарды оның негізгі таралу көзінен бөлу мүмкіндігі;

– қолданушыларға веб-интерфейсті толығымен басқаруға шектеусіз навигация және басқару мүмкіндігі.

1.2 Веб-қосымшаның артықшылықтары

Жергілікті қосымшаларға қарағанда веб-қосымшалардың келесі артықшылықтарын атауға болады:

– қосымшаның қол жетімділігі. Кез келген компьютер ғаламторға қосылу арқылы веб-қосымшаны қолдана алады;

– қолдану қарапайымдылығы. Жергілікті қосымшаларға қарағанда веб-қосымша пайдаланушылардың компьютерлеріне орнатуды қажет етпейді. Қосымшаны өзгерткен кезде барлық қолданушылар өзгертілген нұсқамен бірден жұмыс істей алады;

– веб-технологиялардың және желілік қосылыстардың жоғары деңгейі мен сенімділігі;

– веб-қосымшалар пайдаланушыларына мобильді болуға мүмкіндік береді. Пайдаланушы жұмыс нәтижелерін серверде сақтап, қажет болған жағдайда оларға кез-келген жерден қол жеткізе алады;

– қолданушы бір жүйеден басқа жүйеге ауысқан кезде барлық деректер сақталады.

1.3 Веб-қосымшаның кемшіліктері

Қазіргі таңда қолданушы веб-қосымша арқылы ақпараттамен әртүрлі әрекеттерді іске асыра алады. Негізінен оңдай әрекеттерге ақпаратты алу, жіберу, сақтау немесе өзгерту жатады. Алайда бұл веб-қосымшалардың қолданылуы кең тарағандықтан, олардың көптеген күшті жақтарын атауға болады. Алайда олардың мынандай кемшіліктері де анықталды:

– бұл қосымшаларға ашық қол жетімділік. Веб-қосымшаларға ашық қол жетімді болуына байланысты барлық дерлік веб-қосымшалар олармен жұмыс жасайтын пайдаланушылардың қауіпсіздігін қамтамасыз ете алмайды;

– қолданушы серверге сұрау жіберу барысында сеанс жағдайын қолдау болмауы және әр веб-парақшаны қайта жүктеу кезінде сұраулардың кешігуі;

– веб-желінің негізгі технологиялық архитектурасы және қол жетімді басқару элементтерінің шектеулі жиынтығы жергілікті қосымшаларға қарғанда веб-қосымшалармен әрекеттесуді қиындатады;

– HTTP хаттама ішінде күйлердің сақталмауы;

– клиенттік технологиялардың ұзақ уақыт орындалуы.

1.4 Веб-қосымшалардың қолдану түрлері

Бизнестің алдында тұрған міндеттерге байланысты пайдаланушы қажетті онлайн, яғни веб-қызметті қосымшалар арқылы дамытуға мүмкіндікке ие болады. Веб-қосымшалар арқылы пайдаланушылардың санын көбейтуге болады. Ол арқылы бизнес үлкен қаржылай үлестерге ие болу мүмкіндігі пайда болады. Алайда, ең алдымен қосымшаның желі ішінде танымалдылық пайызы арту керек. Өткені неғұрлым веб-қосымша танымал болған сайын, сол ғұрлым өнімнің сатылу пайызы көбейе түседі. Қазіргі уақытта веб-қосымшаның

қолданылу салалары артқан сайын олардың қолдану түрлері де артады. Олардың негізгі пайдалану түрлері былай қалыптастыруға болады.

Корпоративтік портал бизнес-үрдістерді ыңғайлы және тиімді түрде жүргізуге мүмкіндік беретін көп функцияналды веб-қызмет болып табылады.

Корпоративтік портал арқылы мынандай міндеттерді шешуге болады:

– қолданушыларға қызмет көрсету сапасын жетілдіру;

– қызметкерлердің ұтқырлығын арттыру;

– қажетті құжаттармен қашықтықтан жұмыс істеу мүмкіндігін құрастыру;

– қолданушы әртүрлі қызметтерді онлайн түрде жүзеге асыруы;

– құжаттармен болатын қиын үдерістерді жеңілдету;

– қолданушы және қызметкер арасында болатын үрдістерді автоматтанддыру;

– қызметкерлердің жұмысын жақсарту.

CRM бизнес-үрдістерді басқаруға, жоспарлауға, дамыту жағдайларын тиімді шешуге және қолданушылармен қарым-қатынасты автоматтандыруға арналған құрал болып табылады.

CRM арқылы мынандай міндеттерді шешуге болады:

– тұтынушы базасының қауіпсіздігі мен тұтастығы;

– өнімнің сату пайызы артуы;

– сату бойынша сараптамалардың жүргізілуі.

ERP жүйесі барлық ірі кәсіпорындар үшін жаңа мүмкіндіктер алу үшін қажет.

ERP арқылы мынандай міндеттерді шешу мүмкіндігі бар:

– ақпараттық жүйелерді және есеп беру жүйелерін стандарттау;

– үдерістерді басқару және үндестіру.

1.5 Веб-қосымшаның құру кезеңдері

Веб-қосымшаның құрылу кезеңдері кезең-кезеңмен іске асырылатын әрекеттердің жиынтығы. Қосымша ғаламторда толықтай жұмысын атқару үшін арнайы сараптамалардан және тексерулерден өтуі қажет. Құрылу кезеңі, ең алдымен, маркетингтік жоспарлаудан өтеді. Яғни қосымшаның қызметін анықтау іске асырылады. Қызметтерді анықтау барысында қосымша қолданушыларға қандай пайда әкелу керек және пайдаланушылар үшін қосымша қандай құрамдас бөлігі болып табылады деген мәселелер қарастырылып, жауап беру керек. Веб-қосымшаның қызметтік қолдауы неғұрлым кең болса, соғұрлым қолданушылар саны арта түседі. Қосымшаның қызметін негізгі 6 кезең арқылы іске асыруға болады.

Бірінші кезеңде веб-қосымшаның идеясы және құрылымына назар аударылады. Бұл кезеңде веб-қосымшаның қандай бөлімдерден тұратыны, негізгі міндеттері мен қандай мақсаттарда қолданылуы қарастырылады.

Екінші кезеңде зерттеу үдерісіне жүгінеді. Зерттеу кезінде веб-қосымшаның құны іске асыруға кететін қаржыдан аспауы керек. Қосымшаны

жылдам іске асыру үшін зерттеу кезеңін бесеңдетпеуге болады. Сондықтан зерттеу үрдісі келесі кезеңдерге сәйкес болуы керек:

– зерттеу барысында ең алдымен бәсекелестердің веб-қосымшаларын зерттеу маңызды. Бәсекелестерді зерттеу барысында веб-қосымшаның сыртқы көрісіні емес, қосымшаның негізгі мәзірдегі ақпараттардың мазмұнына және қаңдай бөлімдерге жүктелгеніне назар аудару қажет:

– ұқсас санаттарды түрлендіру;

– ғаламтордағы тенденцияларға жүгіну арқылы веб-қосымшаны бәсекелестерге қарағанда ерекшелендіру.

Үшінші кезең веб-қосымшаның прототипін құрудан тұрады. Прототипті құру үшін мынадай негізгі стандарттық жиынтық қолданылады:

– веб-қосымшадағы басқару элементтері кішкентай түймелер ретінде қолдану;

– навигациялық панелі веб-қосымшаның жоғарғы бөлімінде орналасуы;

– суреттердің негізінен тіктөртбұрыш ретінде қолданылуы;

– екі немесе үш шрифт түрлерін қолдану.

Төртінші кезең барысында веб-қосымшаның мәтіндік мазмұның сұрыптау жүзеге асырылады. Бұл кезеңде веб-қосымшаға қолданылатын суреттер, бейне файлдар, қолданушы қолдануға арналған ақпараттар жиналу арқылы веб-қосымшаның мәтіндік мазмұны құрылады. Мәтін жазу барысында негізгі критерийлерден құрылады:

– компания туралы қысқаша дерек беру;

– компанияның атқаратың қызметтері туралы толық мәлімет жазу;

– компания бәсекелестерімен салыстырып, ерекше 3 негізгі артықшылық туралы мәтіннің болуы;

– компанияның қызметтері мен өнімдердің артықшылықтарына негізделу.

Бесінші кезеңде жобалау арқылы веб-қосымшаны жаңа деңгейге көтеру үрдісі жүргізіледі. Бұл кезең барысында алдынғы кезеңдердің нәтижелерін біріктіру арқылы жаңа ерекше веб-қосымшаның сыртқы келбеттері құрылады. Жобалау кезеңі барысында назар аударатын жағдайлар:

– веб-қосымшаның негізгі бетіне пайдаланушылардың назарын аудару;

– сапасыз суреттерді веб-қосымшада пайдаланбау;

– ерекше шрифт қолдану;

– жаңа және ескі шолғыштарда веб-қосымшаның бейімделу қызметі;

– әртүрлі құрылғыларда бір қалыпты жұмыс істеу.

Алтыншы іске асыру кезеңі. Бұл кезең техникалық үрдістердің әрекеттесу ретінде жасалады. Яғни бұл дегеніміз арнайы әзірлеу және басқару технологиялардың кең қолдануы іске асырылады.

Іске асыру кезеңі ең бірінші, HTML кодтау тілі арқылы веб-қосымшаның қаңқасы құрылады және содан кейін арнайы стильдеу тілі CSS көмегімен браузерлерде қолданушыларға сырттқы келбеті жағымды көріну үшін үлес қосады. Веб-қосымшаның негізгі бөліктері құрылған соң, басқару жүйелеріне

қосуға болады. Басқару жүйелері қолданушылардың іс-әрекеттерін бақылауға мүмкіндік береді.

1.6 Әзірлеу архитектурасы

Қазіргі уақытта ғаламторда веб-қосымшалардың кең таралуына байланысты қосымшалардың қолдану трафигі мен танымалдылығы арта түсуде. Қолданушы қолданатын веб-қосымшаларды толықтай қолдану үдерісі кедергілерсіз жасалу үшін әзірлеу архитектура ұғымы пайданылады. Кез келген қосымшаның әзірлеу архитектурасы екі кезеңнен тұрады. Бірінші кезең клиент жағында жұмыс істейді, ал екінші кезең сервер жағында жұмыс істейді.

Клиент жағында жұмыс істейтін және ақпаратты қашықтағы сервердің көмегіне жүгінетін бағдарлама *клиент-серверлік* бағдарлама деп аталады, ал веб-браузерде толық жұмыс істейтін бағдарлама веб-қосымша ретінде белгілі.

Клиент жағында істейтің ақпаратты алу үшін қашықтан серверге сұрау жасалады. Сұраулар клиенттің бағдарламасы немесе қолданушы интерфейсі арқылы өзара әрекеттесу арқылы сервермен жүзеге асырылады. Веб-қосымшада пайдаланушының өзара әрекеттесуі веб-шолғыш арқылы жүзеге асырылады. Бағдарламалау тіліне байланысты клиенттің қосымшасы кросс-платформа болуы мүмкін. Веб-қосымша платформаға байланысты болмайды, өйткені ол үшін тек қолданушының веб-шолғышы қажет. Кросс-платформалық жүйе – қосымшаны клиенттің операциялық жүйесінің бейімделуі үшін қолданылатын үдеріс.

Қолданушылардың веб-қосымша арқылы жіберген сұрауларын қабылдау үшін серверлік үдеріс кезеңі арқылы сұраулар қабылданады. Сервер жағында жұмыс істеу үдерісі тікелей деректер қоры арқылы өзара байланыс ұйымдастырады. Көптеген сұраулар жіберу кезінде серверге қосымша жүктемелер қосылады. Олардың бәрін толықтай қабылдау үшін серверлік кезеңде қосымша кластер қосылады. Кластер арқылы қолданушы жіберген сұраулар кезек-кезекпен орналастырылып сервердің жылдығын бірнеше рет тездетеді.

1.7 Клиенттік бөлімді әзірлеу технологиялары

1.7.1 HTML

Қазіргі уақытта ғаламтор арқылы жұмыс істейтін веб-технологиялардың көп бөлігінде HTML қолданылады. HTML (Hyper Text Markup Language) бұл гипермәтіндік құжаттарды жасауға, гипермәтіндік құжаттартар арасында байланыстарды ұйымдастыруға және ақпаратты әртүрлі элементтер арқылы түрлендіруге мүмкіндік беретін әзірлеу құралы. Барлық шолғыштар HTML тілін негізгі стандарт ретінде қарастырады. Шолғыштарда ақпараттармен алмасу үшін арнайы ғаламторлық сервистерді қолданылады. Қазіргі уақытта HTML және сервистер арасында ақпараттармен алмасу үшін гипермәтінді ақпаратты тарату хаттамасы яғни HTTP (Hyper Text Transfer Protocol) және

қауіпсіз гипермәтінді ақпаратты тарату хатамасы HTTPS (Hyper Text Transfer Protocol Secure) технологиялары арқылы іске асырылады. Бұл екі хаттама қолданушы серверлерге сұраулар жіберу кезінен бастап сервер қолданушыға нәтижені жіберу кезеңіне дейін жұмыс жасайды. Хаттамалардың айырмашылығы HTTPS жеке хаттама ретінде жасалып 443 портта жұмыс істейді және арнайы криптографиялық хаттаманы қондырмасын қолдану арқылы қолданушының ақпаратың қауіпсіз түрде жіберіледі. HTTPS хаттамасы SSL және TSL криптографиялық хаттаманы қолданғандықтан шолғыштар көбінде қолданушыларға қауіпсіз ретінде хабарланады. Бұл хаттама көбінесі онлайн-төлемдері бар веб-қосымшаларда қолданылады. Мысалы: онлайн дүкендерде, билеттерді брондау үшін, ақша аудару немесе пайдаланушы құпия ақпараттарды енгізу және алуда жұмыс жасайды.

Барлық шолғыштар HTML құжаттарын кедергілерсіз анықтау үшін арнайы қосылу тегтері қолданылады. Қосылу тегтері екі негізгі элементтен құрылады. Бірінші қосылу тег <!DOCTYPE html> және екіншісі <html>. Екінші тегтің ішіне lang ақпаратын қосуға мүмкіндік бар. Яғни ол арқылы веб-қосымшаның қандай тілде бейімделгені анықталады. Бұл тегтер келесі түрде кездеседі:

```
<!DOCTYPE html>
<html lang="ru">
</html>
```

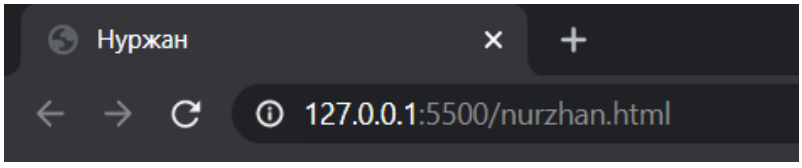
Кез келген HTML құжатты негізгі екі бөліктен құрылады. Ол бөліктер <html> тегтің ішінде жіктеледі, яғни құжаттың басы және денесі. Құжаттың басы <head> тегі арқылы, ал денесі <body> арқылы сипатталады.

Құжаттың басы бірнеше қосымша тегтерден құрылады. Олар негізінде веб-қосымшаның тақырыптық мәнің сипаттауға және басқа құжаттарды қосуға болатын мүмкіндіктерін береді.

Басқа құжаттарды қосу үшін <link rel="stylesheet" href="бөгде құжат"> арқылы іске асырылады. <link> тегі құжаттын басында бірнеше рет кездесуі мүмкін. Бұл тег href атрибуты арқылы басқа құжаттардың мекен-жайын жазу арқылы жұмыс істейді.

Веб-қосымшаның тақырыптық мәнің көрсету үшін <title> тегі қосылады. Бұл тегтін HTML құжатта атқаратын қызметі үлкен. Өйткені осы тег арқылы қолданушылар қандай веб-қосымшаны қолданып жатқаны және ғаламторда іздеу жүйелері сәйкес тақырыптық мәні бар веб-қосымшаларды іздеу жұмысын жылдамдатады. Жазылу нұсқасы:

```
<title>Нуржан</title>
<link rel="stylesheet" href="nurzhan.css">
```

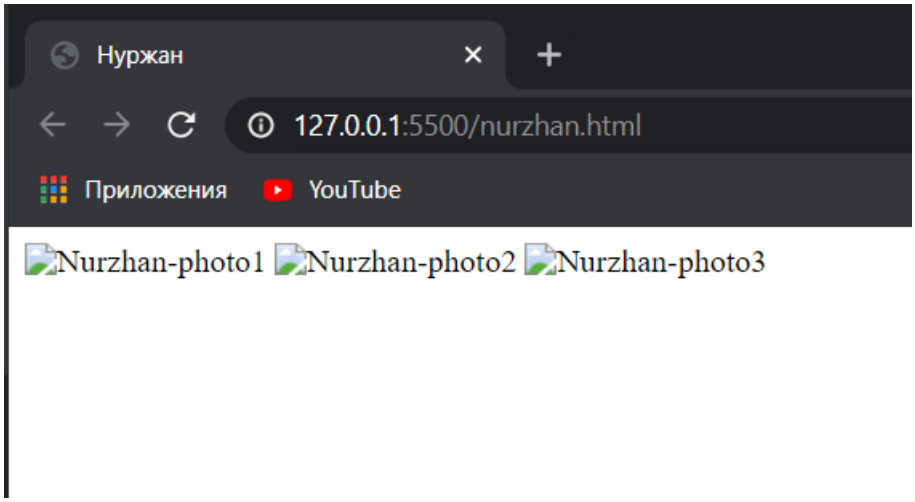


1.1 сурет – Құжаттың тақырыбы

HTML құжаттың ең ауқымды бөлігі ол - <body> тегі немесе құжаттың денесі деп айтуға болады. Құжаттың денесінде элементтердің пішіндерін тегтер арқылы өзгертуге құжаттар арасында болатын байланысты қосу, әртүрлі графикалық суреттерді еңгізу, видео аудио құжаттарын қосуға және басқада мүмкіндіктерге ие бола алады.

Құжаттың денесі екі түрлі тегтерден құрылады. Бірінші түрі жұпсыз ол – ашық тегтер, екінші түрі жұппен жазылатын, яғни ашық және жабық тегтер. Ашық тегтер қатарына суреттер салуға мүмкіндік беретін тег <img> кіргізуге болады. <img> тегі бірнеше параметрлерді қабылдайды. Олар src және alt. Src ішіне суреттің компьютерде немесе серверде орналасу мекен-жайын жазу арқылы құжаттың денесіне түседі. Ал егер ол суреттің орналасу мекен-жайы дұрыс жазылмаған жағдайда немесе ол сурет серверде және компьютерде жойылып кеткен жағдайда қосымша alt параметрі іске қосылады. Alt параметрінің негізгі жұмыс істеу механизмі суретті сипаттайтын мәтінді қолданушыларға көрсету арқылы жұмыс істейді. Жазылу нұсқасы:

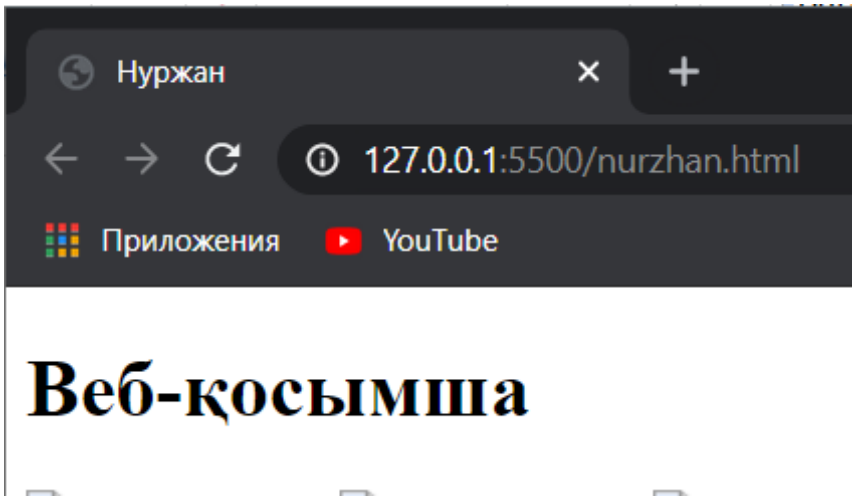
<img src="/img/nurzhan.jpg" alt="Nurzhan-photo1">
<img src="/img/nurzhan.png" alt="Nurzhan-photo2">
<img src="/img/nurzhan.gif" alt="Nurzhan-photo3">



1.2 сурет – Alt параметрінің көрінісі

HTML құжаттардың ішінде әртүрлі мәтіндер болғандықтан, оларды түрлердіру үшін арнайы тегтерді қолданады. Негізгі тегтер:

<h1>Веб-қосымша</h1> - мәтінді үлкен және қалын түрге келтіреді. Ғаламторда іздеу жүйелері осы тег арқылы ақпаратты қолданушыға нәтиже ретінде шығарады.



1.3 сурет – H1 тегі арқылы құжатқа мәтін еңгізу

<p> - ауқымды мәтіндерге арналған және құжаттағы мәтінге абзац беруге арналған тег.

Нуржан - мәтіннің ең маңызды бөліктерін сипаттауға арналған тег.

Әрбір тегтің атқаратын қызметі болады. Бірақ жалпы құжаттың денесінде үш тегтің атқаратын қызметі ауқымды түрде жұмыс істейді.

Бірінші тег - . Бұл тег құжаттың денесінде контейнер ретінде html құжаттың элементтерін блоктарға бөледі:

```
<div class="nurzhan-block">
    <div class="nurzhan-item"></div>
    <div class="nurzhan-item"></div>
    <div class="nurzhan-item"></div>
</div>
```

Екінші тег - <form>. Бұл тегтің негізгі атқаратын қызметі форманы жасау. Форма сервермен тікелей жұмыс істейді. Тегтің ішіндегі ақпаратты сервер өңдеу арқылы деректер қорына жіберіледі. Жазылу нұсқасы:

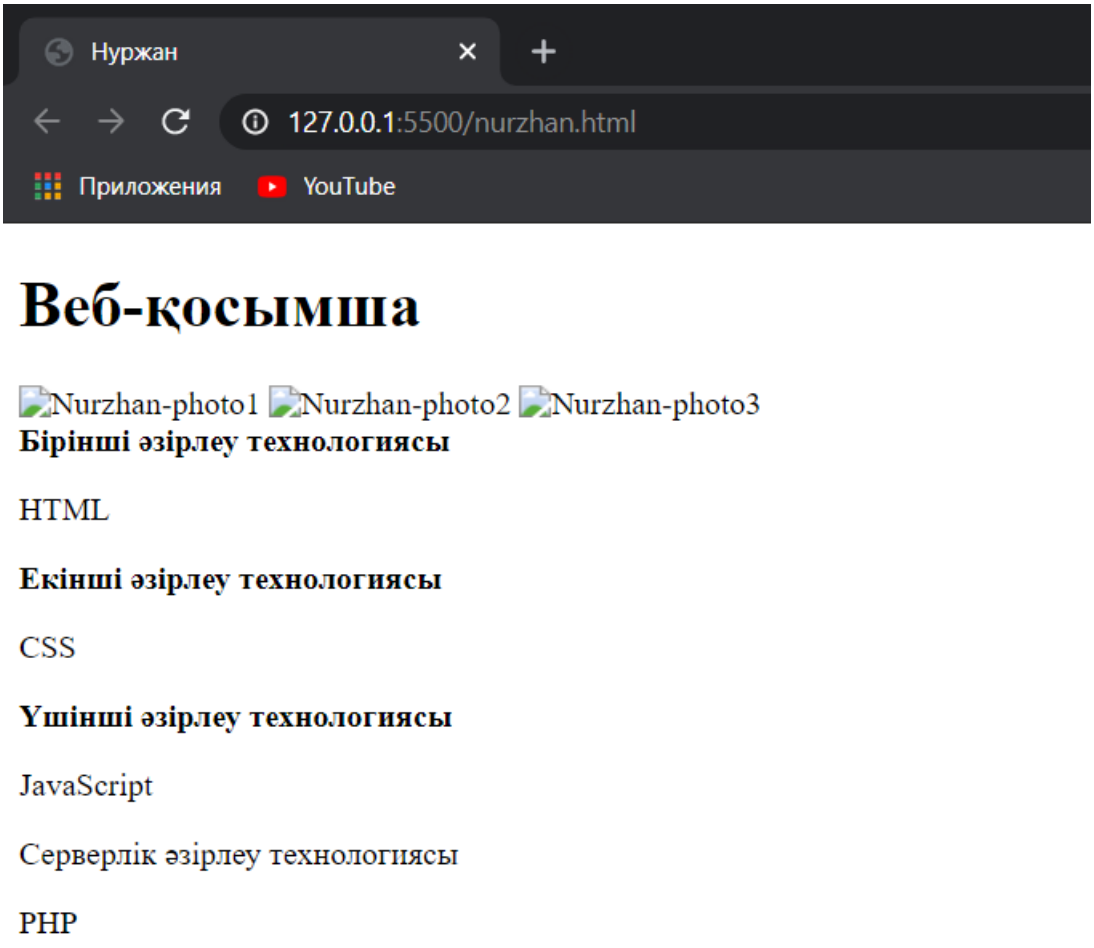
```
<form>
    Ақпарат формасы
</form>
```

Үшінші тег - <a>. Қолданушы басқа html құжаттарға өту үшін <a> тегі қолданылады. Бұл тег href параметрін қабылдау арқылы басқа сілтемелерге қолданушыны басқа құжаттарға апарады.

Құжаттың құрылымын бұтақ тәріздес қалыпқа келтірген құжат дұрыс әзірлену үрдісі ретінде қабылданады.Жоғарыда келтірілген тегтерді қолдана отырып қарапайым html құжат құруға болады:

```
<!DOCTYPE html>
<html lang="kz">
<head>
    <meta charset="UTF-8">
    <title>Нуржан</title>
    <link rel="stylesheet" href="nurzhan.css">
```

```
</head>
<body>
    <h1>Веб-қосымша</h1>
    <div class="nurzhan-photo">
        <img src="/img/nurzhan.jpg" alt="Nurzhan-photo1">
        <img src="/img/nurzhan.png" alt="Nurzhan-photo2">
        <img src="/img/nurzhan.gif" alt="Nurzhan-photo3">
    </div>
    <strong>Бірінші әзірлеу технологиясы</strong>
    <p>HTML</p>
    <strong>Екінші әзірлеу технологиясы</strong>
    <p>CSS</p>
    <strong>Үшінші әзірлеу технологиясы</strong>
    <p>JavaScript</p>
    <div>Серверлік әзірлеу технологиясы</div>
    <p>PHP</p>
</body>
</html>
```



1.4 сурет – Қарапайым веб-қосымшаның бастапқы беті

1.7.2 CSS

Веб-қосымшалардың сыртқы интерфейсі қолданушыларға сәнді көріну үшін CSS (Cascading Style Sheets – яғни каскадты кестені стильдеу) тілін қолданылады.

CSS тілі арқылы веб-қосымшаның ішіндегі элементтерді сыртқы көрінісін безендіруге негізделеді. Бұл каскадты кестені стильдеу тілі арқылы веб-қосымшалардың элементтері пішінің өзгертуге, элементтердің түсін беруге, сондай-ақ анимация беруге мүмкіндіктері бар.

Басқа құжаттарға қосылу кезінде ең дұрыс қосылу түрі ,<head> тегі ішінде <link> арқылы қосылу.

```
<head>
        <title>Нуржан Веб-қосымша</title>
        <link rel="stylesheet" href="nurzhan.css">
</head>
```

CSS тікелей HTML құжатты арқылы жұмыс жасайды. HTML құжаттағы элементтерді каскадты кестені стильдеу тілі арқылы стильдеу үшін класстарға сураулар жіберіледі. Яғни нүкте арқылы басқа құжаттағы элементтеріне сұраулар жіберіледі:

```
.nurzhan-block{
    color: orange;
}
```

Бұл жерде нүкте арқылы HTML құжаттың тегіне байланыс ұйымдастырылады, нүктеден кейін элементтің аты жазылып, керек кескіндер жазылады. Color арқылы элементтердің түсі беріледі.

Каскадты кестені стильдеу тілінде пішінді сипаттауға арналға функциялар бар. Оларға:

width – құжаттағы элементтердің енің беруге арналған;

height – құжаттағы элементтердің биіктігін беруге арналған;

Функциялардың есептеуге арналған бірнеше өлшем бірліктері болады. Бірінші өлшем бірлігі “px”; Бұл өлшем бірлігі кең таралғандықтан көптеген жобаларда қолданылады. Екінші өлшем бірлігін пайыз арқылы беруге болады. Бірақ пайызбен берілген өлшем бірлік веб-қосымшаның элементтеріне керішінше әсерін беру мүмкін. Пиксельмен және пайызбен берілген өлшем бірліктердің айырмашылығы, пайызбен берілген өлшем бірлікті ескі браузерлер қолдау мүмкіншіліктері болмайды.

Пиксельмен беру тәсілі:

```
.nurzhan-block {
        width: 24px;
        height: 24px;
}
```

Пайызбен беру тәсілі:

```
.nurzhan-block {
        width: 20%;
```

```
        height: 20%;
}
```

Веб-қосымшаның элементтеріне динамикалық қызметті қосуға арналған анимациялық функциялар болады. Ол қызметті қосу барысында @keyframes тәсіліне анимацияның аты жазылады. Бұл тәсіл арқылы CSS құжат анимацияны тіркеу механизмі іске асарылады. Анимация беру кезінде ең алдымен CSS құжаттың ішіне бастапқы нүктесін бекітіп, аяқталатын соңғы нүктені белгідеу қажет. Бастапқы нүкте from, ал соңғы нүкте to арқылы белгіленеді:

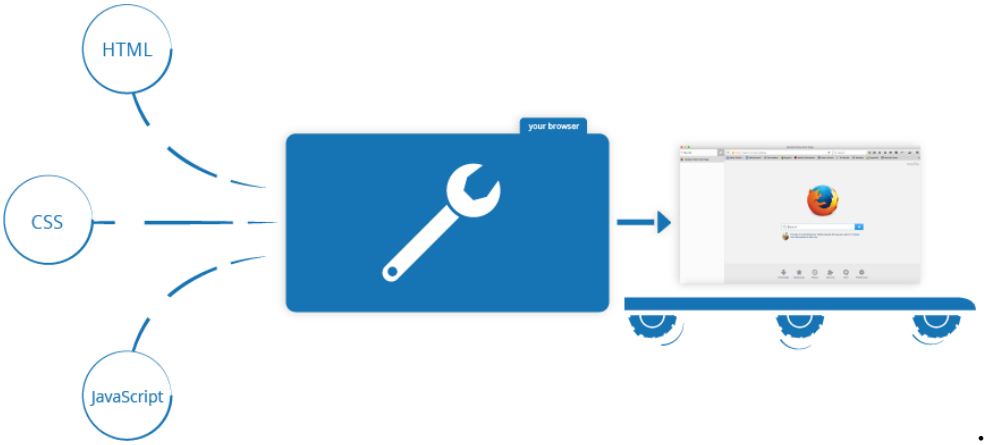
```
<!DOCTYPE html>
<html>
 <head>
  <meta charset="utf-8">
  <title>Веб-қосымшаға анимация</title>
  <style>
   .nurzhan-image {
    animation-name: nurzhan-image;
    animation-duration: 0.2s;
   }
   @keyframes nurzhan-image{
    from {
     opacity: 0;
     transform: translateY(0%);
    }
    to {
     opacity: 1;
     transform: translateY(10%);
    }
   }
  </style>
 </head>
 <body>
  <img src="/img/nurzhan.jpg" class="nurzhan-image">
 </body>
</html>
```

Жоғарыда көрсетілгендей, ең алдымен @keyframes арқылы nurzhan-image анимациясы CSS құжатқа тіркелу үдерісі жүргізіледі. Тіркелу үдерісі толық орындалғаннан кейін бастапқы нүктеге элементтің стильдері беріледі. Стильдер беру кезінде элементтердің қимыл-қозғалыстары сипатталады. Бастапқы нүкте толық орындалғаннан кейін, аяқталатын нүкте үдерісі орындалады.

1.7.3 Javascript

Javascript бағдарламалау кодтау тілі арқылы веб Ол қосымшадағы элементтерді басқару, қолданушы қолданатын интерфейспен өзара әрекеттесу және басқаруға мүмкіндіктерін береді. Бұл тіл арқылы веб-қосымшаның мынадай динамикалық әрекеттерді қосуға болады. деректерді көрсету, анимациялық іс-әрекеттер, ақпаратты өңдеу және тағы басқад а мүмкіндіктер.

Javascript көмегімен веб-парақта кез келген операцияларды орындауға болады. Javascript көп жағдайда қолданушы веб-браузермен әрекеттесу кезінде қолданылады. Мысалы: қолданушы батырманы басқан кезде шолғышта арнайы терезені шақыру үдерісі Javascript арқылы орындалады. Браузер HTML және CSS кодтарын өңдеу үдерісі аяқталған мезетте, Javascript коды іске қосылады.



1.5 сурет – Javascript веб-шолғышта өңделу үдерісі

Әдетте, веб-шолғыш Javascript кодын өңдеу үдерісі кезінде жоғарыдан төмен қарай орындалады. Сондықтан код жазу барысында рет-ретімен жазуы қажет. Javascript құжатты қосу үдерісі HTML құжатта іске асырылады. Яғни, ол <script> тегі арқылы іске асырылады. Жазылу реті келесі түрде кездеседі:

<script src="myscripr.js "></script>

Құжатты қосудан басқа <script> тегі кодты HTML құжатта орындауға мүмкіндік береді. Осындай тәсілді көбінесе Javascript коды кішкентай көлемдегі бағдарламаларды жазуға арналады. Жазылу реті келесі түрде кездеседі:

```
<!DOCTYPE html>
<html lang="kz">
<head>
   <meta charset="UTF-8">
   <title>Нуржан</title>
   <link rel="stylesheet" href="nurzhan.css">
</head>
<body>
<script>
alert("Терезеге хабарлама көрсету.");
</script>
```

</body>
</html>

Жоғарыда көрсетілгендей, қарапайым бағдарламаларды орындау үшін <script> тегі қолданылады. Бұл тегтің ішінде alert тәсілі арқылы қолданушыға хабарлама веб-браузерде көрсетіледі. Осындай тәсілдердің көмегімен бағдарламаның орындалу жылдамдығы арта түседі. Javascript құжатты сырты құжаттардан айыру үшін “js” форматта сақтау керек.

Javascript бағдарламалау тілінің ерекшеліктері:

– кез келген тегке код арқылы жылдам еңгізу;
– өзара сервермен әрекеттесу;
– қолданушылардың деректерімен жұмыс істеу;
– құжаттағы элементтердің сыртқы келбеттерін анықтау.

Бұл бағдарламалау тілі қолданушының веб-парақша тегінде кез келген іс-әрекетті істеуге мүмкіндік береді. Өйткені бұл бағдарламалау тілі бастапқыда веб-шолғыштармен жұмыс жасау құралы ретінде ойластырылған. Қазіргі кезде веб әзірлеушілер осы бағдарламалау тілін кең қолданғандықтан, көптеген қосымша әзірлеу құралдары пайда болып жатыр. Қосымша құралдар арқылы әзірлеу жұмысын жеңілдету, қысқарту, болашақта қосымшаны қолдауға мүмкіндіктер береді.

1.8 Серверлік бөлімді әзірлеу технологиялары

1.8.1 PHP

Қазіргі кезде веб қосымшаның серверлік бөлімін әзірлеу технологиясына PHP бағдарламалау тілі қолданылады. Бұл бағдарламалау тілі басқа веб бағдарламалау тілдер арысында ең танымал тілдерінің қатарында жатыр. PHP скрипттерді жазу арқылы жұмыс істейді. Скрипттіні жазу арқылы қолданушы жасайтын көптеген тапсырмаларды автоматтандыруға мүмкіндік пайда болады. Бағдарламалау арқылы қолданушының уақытын үнемдеуге және басқа процестерге қауіп қатерлерді болдырмайды. PHP бағдарламалау тілі бастапқыда веб-әзірлеу үшін пайданылған, сондықтан да веб-қосымшада серверлік жағын әзірлеу үшін кең қолданылады. Яғни веб-қосымшада көптеген динамикалық парақшалар және кішкентай веб-бағдарламаларды жазуға болады. Мысалы осы бағдарламалау тілің былай қолдануға болады:

– веб-қосымшада деректер қорымен жұмыс істеу;
– қолданушыларға пікірлерді жазу жүйесін құру;
– веб-қосымшаның іздеу жүйесін әзірлеу;
– тіркелу парақшасын және кіру парақшасын жасау.

Бұл бағдарламалау тілінің ерекшеліктері, кемшіліктері және артықшылықтары болады.

Ерекшеліктері – бұл бағдарламалау тілінің ішінде динамикалық типизацияның нашарлығы. Яғни бұл дегеніміз, бағдарламаны орындау кезінде кез келген айнымалылардың түрлері анықталады, бағдарламаның орындау сәтінде типтерді әртүрлі қолдануға болады. Осындай ерекшеліктеріне сай

бағдарлама бір жағынан жылдамырақ жұмыс жасайды, жақсы және икемді оқылады, ал екінші жағынан қателіктердің болу ықтималдылығы жоғарылайды. Бұл қателіктерді тек бағдарламаны іске қосқанда ғана байқауға болады.

Артықшылықтарына тоқталатын болсақ, бұл бағдарламалау тілі тегін алуға болады. Әзірлеушілердің жұмысын жеңілдету үшін көптеген қосымша кітапханалар деректері құрылған. Көптеген серверлермен үйлесімді және икемді жұмыс істейді. Бұл бағдарламалау тілін басқа серверлік тілдерге қарағанда үйрену оңайға түседі. Қарапайым түрде жазылады және кластар мен нысандармен жұмыс істеу жүйесі жақсы. Мысалы осылай қарапайым қодтау мүмкіншілігі көрсетіледі:

```
<!DOCTYPE html>
<html lang="kz">
<head>
   <meta charset="UTF-8">
   <title>Нуржан</title>
   <link rel="stylesheet" href="nurzhan.css">
</head>
<body>
<?php
echo “Нуржан әзірлеуші”;
</body>
</html>
```

Кемшіліктеріне айтатын болсақ, PHP бағдарламалау тілімен әрекеттесу үшін гипермәтіндік құжаттармен жұмыс істейтін тіл яғни, HTML және каскадты кестені стильдеу тілі CSS білу қажет. Кейбір жағдайларда Javascript бағдарламалау тілі де бұл серверлік бағдарламалау тілімен әрекеттеседі. PHP бағдарламалау тілі оңай және икемді болғандықтан әзірлеушілер сапасы төмен код жазады. Сол себебтен қателесу жеңіл және сол қатені табу қиындайды. Сондықтан бұл бағдарламалау тілін толықтай дұрыс қолдану үшін қодтау сапасына назар аудару қажет. Қосымша кітапханалардың әртүрлі стильде жазылуы кесірінен әзірлеушілердің бағдарлама жасаудың жылдамдығы баяулайды.

Қауіпсіздік мәселесін де кемшіліктер қатарына қосуға болады. Өйткені бұл бағдарламалау тілі тегін және ашық түрде қол жетімді болғандықтан, басқа әзірлеушілер қатерлі баг кодтарын еңгізу арқылы өз мақсаттарында пайдалануы мүмкін. Тағы жиі кездесетін кемшілігі ол: бағдарламалау тілі оңай болғандықтан әзірлеушілер кодты нашар сапада жазуы және оны болашақта басқа құрылғыларда қолдау қиындығы.

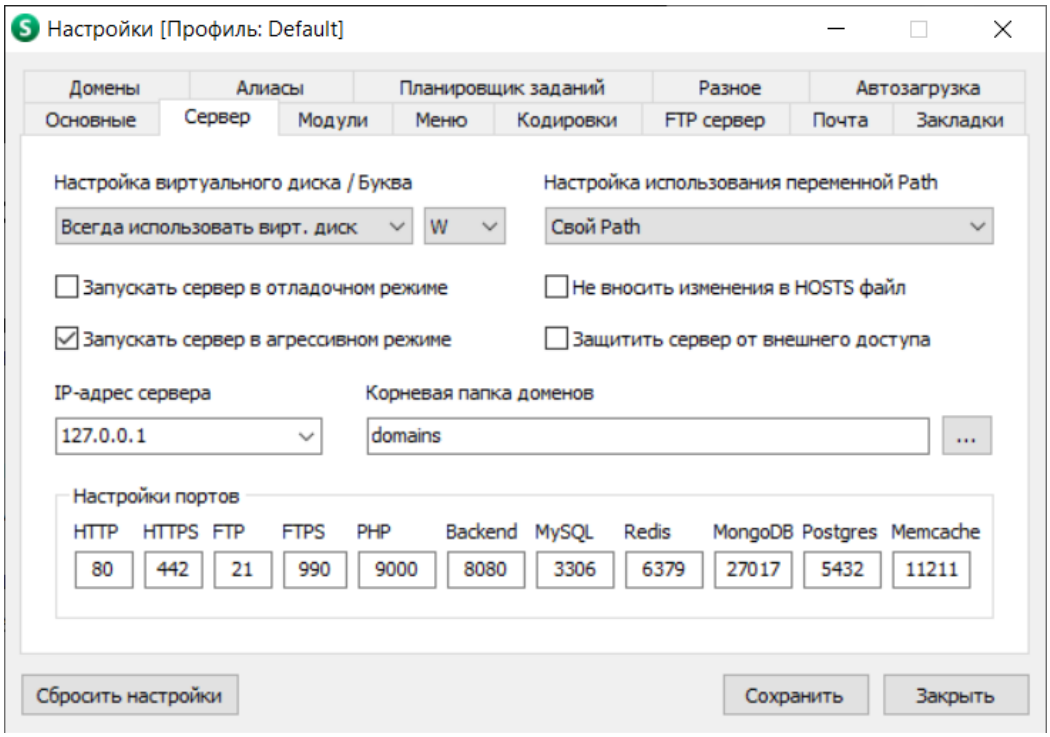
1.8.2 Open-server

Әзірлеу кезінде веб-парақшалардың жұмыс нәтижесін қарау үшін серверлік бағдарлама қолданылады. Серверлік бағдарламалар екі түрлі жұмыс

істейді. Бірінші түрі серверлік бағдарламалау тілі PHP арқылы ғаламтордағы серверге құжаттарды салу тәсілімен. Екінші түрі жергілікті сервер арқылы жұмыс істеу болып табылады. Осы екі түрдің артықшылықтары болады. Жергілікті сервер ғаламтордағы серверге қарағанда жылдамырақ істейді. Егер қолданушының байланыс жүйесі нашар болған жағдайда, ғаламтордағы сервермен жұмыс істеу қиындайды. Сондықтан әзірлеушілер жасалған қосымшаны сынау үшін жергілікті серверді қолданады. Ондай жергілікті серверлік бағдарламаның бірі Open Server.

Open Server тегін бағдарламалар қатарына жатады. Бұл бағдарламаның үш түрі болады: қарапайым қолданушыға арналған, кеңейтілген түрі, көпфункционалды. Қолданушы немесе әзірлеуші бұл бағдарламаны әртүрлі мақсатта қолданғандықтан осылай бөлінеді. Бағдарлама жиі жаңартылуды қажет етеді. Өйткені әрбір жаңа нұсқаларда жиі кездескен қателер түзетіліп, жаңа түрде жұмыс істеуге мүмкіндіктер пайда болады. Open Server шағын бағдарлама болғандықтан, кейбір жағдайда бұл бағдарламаны орнатусыз флешка арқылы іске қосуға болады.

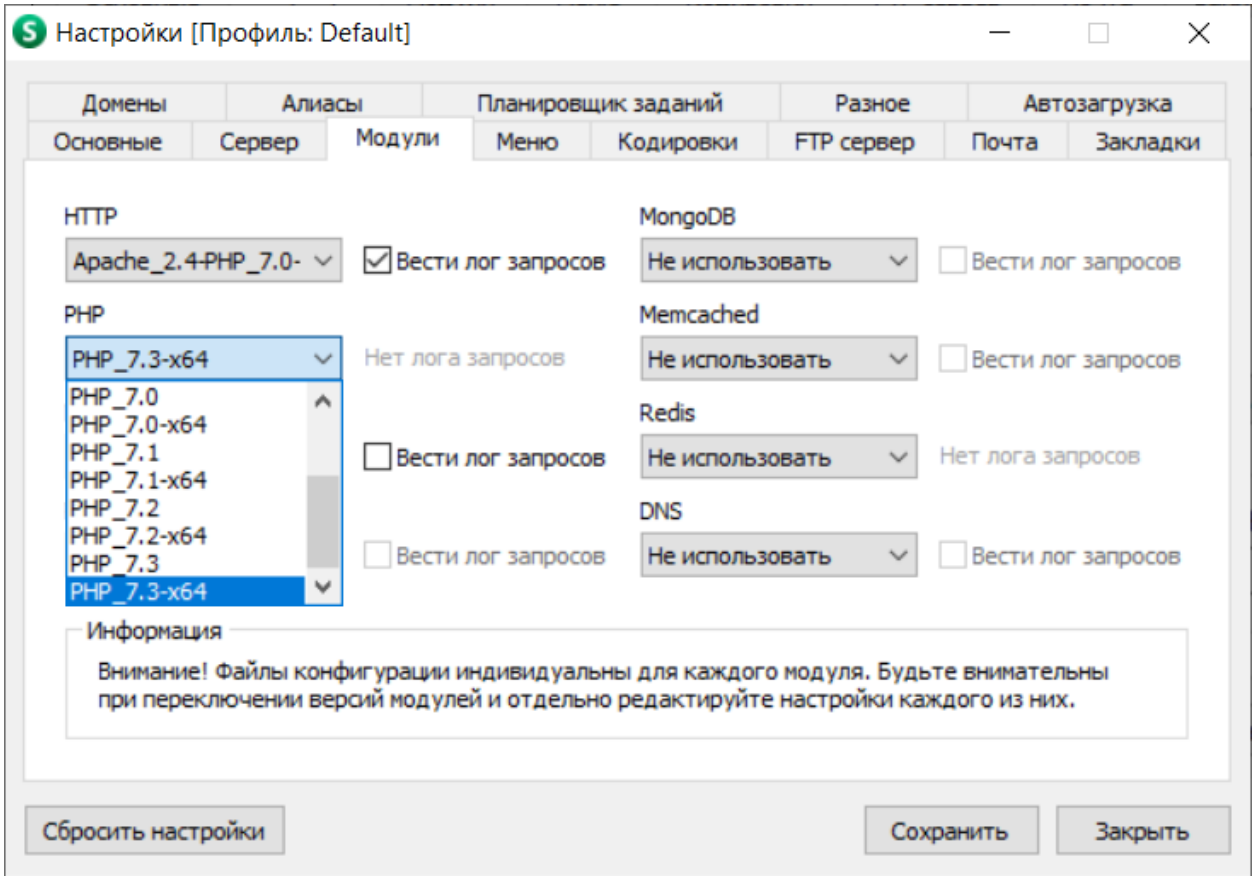
Open Server жұмыс жасау алдында қолданушыға немесе әзірлеушіге сай параметрлерін қосу қажет. Әзірлеуші құжаттардың орналасу орның өзіне керек мекен-жайын жаза алады. Егер порттардын нөмірі басқа бағдарлама қолданып жатқан жағдайда, әзірлеуші Open Server арқылы сол порттың нөмірін оңай ауыстыра алады.



1.6 сурет – Бастапқы параметрлері

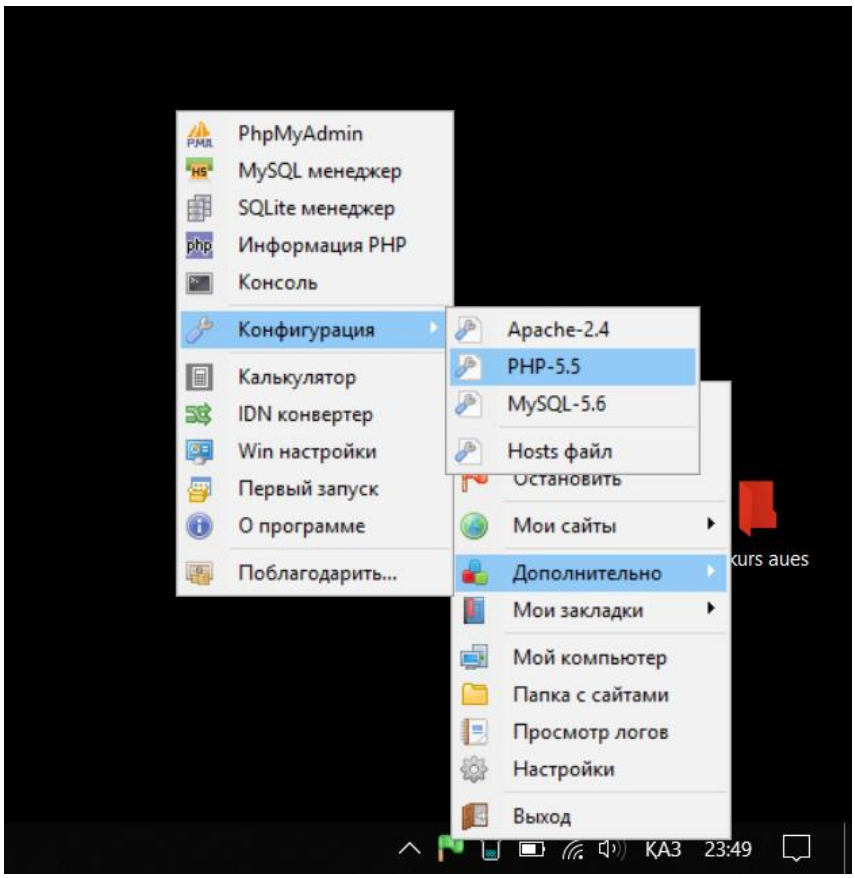
Әзірлеу кезеңі барысында қолданылатын әзірлеу құралдары арасында бір-бірімен түйісуі мүмкін. Өйткені әр құралдардың өзіндік нұсқалары болады. Соған байланысты әзірлеу барысында көптеген қателіктердің шығу қауіпі пайда болады. Ондай қателіктер болмау үшін бұл жергілікті серверде әзірлеуші

өзіне керек нұсқаларды таңдауға болатын мүмкіндік қосылған. Бірақ бұл тәсіл толықтай орындалу үшін ең алдымен керек нұсқаны жүктеп алу керек.



1.7 сурет – Нұсқаларды таңдау мүмкіндігі

Қосу барысында әкімші атынан іске қосуды қажет етеді. Әзірлеуші кейбір жағдайларда әзірлеу құралдары ішіне өзіндік параметрлерді еңгізу мәселелері туындауы мүмкін. Бұл жағдайда жергілікті сервер толықтай әзірлеу құралдарының параметрлерін өзгертуге мүмкіндік береді.



1.8 сурет – Әзірлеу құралдарының параметрлері

Көптеген пайдаланушылар осы жергілікті сервер Open Server басқа жергілікті серверлерге қарағанда көп қолданады. Себебі бұл жергілікті сервер ғаламтор беткейінде жаңа қалыптасқан бағдарлама. Тағы бір ерекшелігі бұл жергілікті сервердің үлкен қауымдастығының ұйымдастырылуы.

1.8.3 MySQL веб-қосымшаға қосу

Қазіргі таңда кез-келген веб-қосымшалар деректер қорымен жұмыс істейді. Веб-қосымшалардың негізгі мақсаты қолданушыға қызмет көрсету арқылы қаржылай төлемдерді алу. Ол қызметтерді кім қолданғаның сақтап отыру үшін деректер қоры ойлап табылған болатын. Жобаға байланысты әзірлеушілер әртүрлі деректер қорын қолдану мүмкін. Бірақ көп жағдайда кездесетін деректер қорына MySQL-ді жатқызуға болады.

MySQL деректер қорын басқару жүйесі әлемдегі ең танымал, жұмыс істеу жылдамдығы тез және сенімді деректор қорына жатады. Болашақта веб-қосымшаның таралуы үшін осы деректер қоры жақсы таңдау ретінде есептеуге болады. Бұл деректер қоры әртүрлі операциялық жүйелерде және танымал басқару жүйелерінде сенімді түрде жұмыс істей алады.

Осы деректер қорының сипаттамаларын атауға болады:

– үлкен қауымдастық. Қауымдастық арқылы қателіктерді тез және икемді шешуге болады;

– сенімді және тұрақты жұмыс істеуі. Сенімді және тұрақты жұмыс істеу арқылы деректер қорын болашақта жобаны қолдау кезеңі оңай өтеді;

– басқа деректер қорынан ең жақсы жерлерін қолдану. Яғни, әзірлеушілерге осы деректер қорымен жағымды жұмыс істеу үшін басқа деректер қорларынан ыңғайлы жақтарын қолданады;

– жаңартулардың жиі жасалуы. Бұл арқылы алдынғы нұсқалардағы қателіктерді болдырмауға мүмкіндіг пайда болды;

– үлкен веб-қосымшада жеңіл және таралу жүйесі оңай жасалған.

Жылдамдығы және сенімділігі туралы бойынша: SQL-ден әртүрлі функциялардан бас тарту арқылы осы деректер қоры жылдамдық және сенімділік жағынан басымдылыққа ие болады. Бұл деректер қорының жылдамдығы оқу операцияларында анық байқалады. Бірақ веб-қосымшада үлкен жүктеме арқылы күрделі сұрауларды орындау кажет болған жағдайда, қосымша параметрлерді орнатуды қажет етеді.

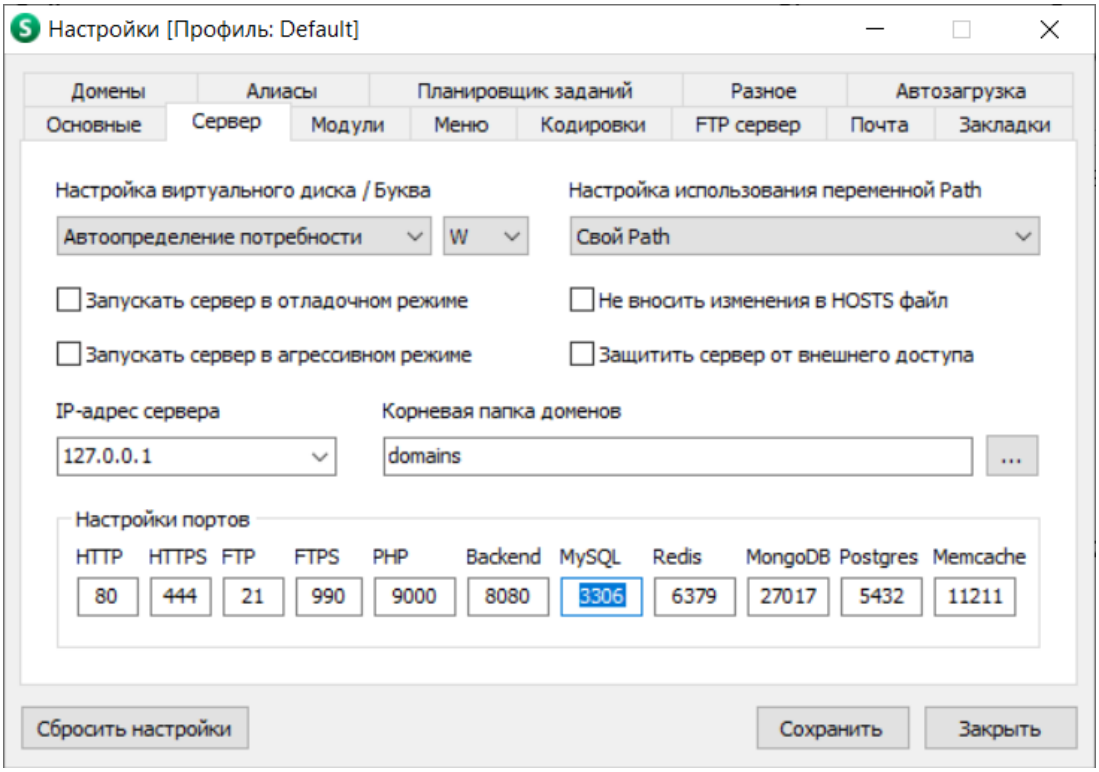
Қарапайымдылығы және танымалдылығы бойынша: MySQL деректер қорының басқаларына қарағанда оңай теңшеуленеді. Көптеген әзірлеушілер осы деректер қорына мән беру себебі, осы қарапайымдылығына сүйенеді.Сонымен қатар бұл деректер қорында басқаларына қарағанда ыңғайлы графикалық интерфейстің болуы.

Бұл деректер қорының танымалдылығына байланысты көптеген веб-қосымшалар осы деректер қорын жобаларға еңгізуі арта түсті. Осыған байланысты осы деректер қоры даму жылдамдығы бірнеше есе ұлғая түсті. Маштабты таралуы туралы айтатын болсақ, бұл деректер қоры бұлтты сервистерде өзін бірқалыпты сенімді ұстайды.

MySQL деректер қорын веб-қосымшаға қосу үшін PHP тілі арқылы деректер қорының мекен-жайы, порт нөмірі және кестенің атын жазу қажет:

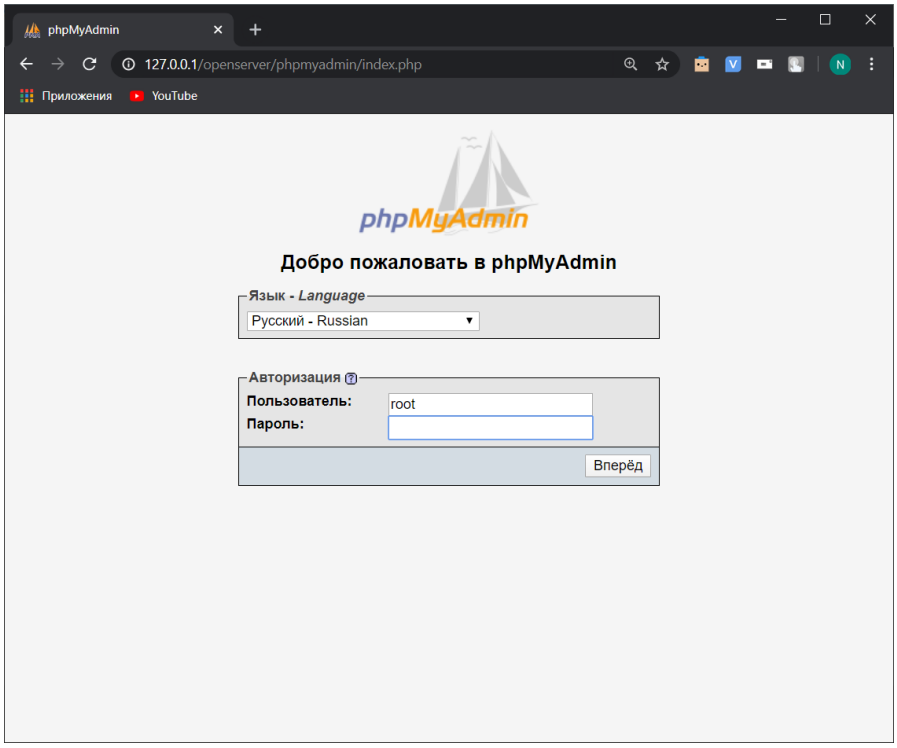
```
<?php
require "libs/rb.php";
R::setup( 'mysql:host=127.0.0.1;port=3306;dbname=nur','root');
session_start();
?>
```

Жоғарыда көрсетілгендей порт нөмірін 3306-ге тең. Бұл деректер қоры толықтай жұмыс істеу үшін осы нөмірді жергілікті серверде көрсету қажет.



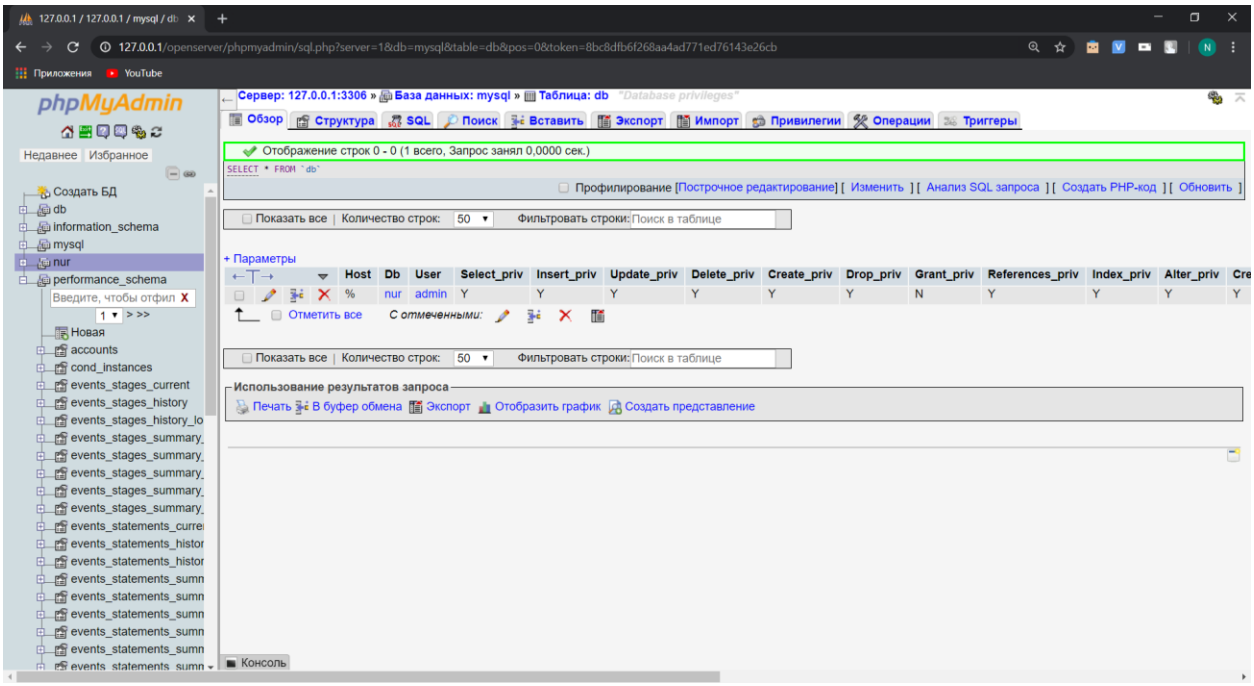
1.9 сурет – Жергілікті серверді деректер қорына қосу үдерісі

Деректер қорын жергілікті серверге қосу үдерісінен кейін, әзірлеуші толықтай жұмыс істеу мүмкіндігіне ие болады. Ең алдымен әзірлеуші жергілікті серверде орналасқан деректер қорына арнайы жасалған графикалық интерфейске кіру қажет. Интерфейс арқылы деректер қорының ішіне әкімші немесе басқа қолданушының атынан кіруге болады.



1.10 сурет – Деректер қорына кіру үдерісі

Бұл деректер кіру үдерісі аяқталғаннан кейін, жергілікті сервер деректер қорына өзіндік қосымша құжаттарды жүктеу кезеңі іске асырылады. Бұл қосымша құжаттар деректер қорының толықтай жұмыс істеуіне және әзірлеу барысында қателіктердің санын азайтуға мүмкіншіліктер береді.



1.11 сурет – Деректер қорының ішкі интерфейсі

Жоғарыда көрсетілген интерфейс арқылы қарапайым кестелерді құруға болады. Егер күрделі үдерістерді жасау керек болған жағдайда, тікелей осы деректер қорына сұраулар немесе кодты енгізу мүмкіншіліктері бар.

2 Веб-қосымшаларға жасалатын негізгі шабуылдар түрлері. Веб-қосымшаның қауіпсіздігі және шабуылдан қорғану мәселесін қарастыру.

2.1 Қауіпті кодты енгізу

Қазіргі уақытта веб-қосымшалардың осалдығын пайдалану арқылы жасалатын шабуылдың ең көп тараған түрі - ғаламтор аралық скриптинг. Ғаламтор аралық скриптинг күрделі жобаларды және тұрақты веб-қосымшаларды сынау барасында күрделі өткізілмегендіктен көптеген зардаптарды тигізеді.

Қауіпті кодты енгізу сияқты шабуыл түрі пайдаланушының шолғышында орындалатын кодты серверге жіберу арқылы іске асырылады. Яғни бұл код клиенттік әзірлеу құралдары HTML гипермәтіндік құжаттарды жасау тілі және Javascript бағдарламалау тілі арқылы жасалады.

Жіберілген қауіпті код сервердің осалдығын пайдалана отырып, қолданушының шолғышы арқылы қол жетімді маңызды деректерді оқып қарауға, өзгертуге және басқаларға жіберуге немесе жүктеуге мүмкіндік алады. Шабуыл жасалған қолданушының шолғышында өзі туралы ақпараттар сақталады. Ол ақпарат cookie құжатты ретінде серверге жіберіледі. Егер зиянкес шабуылды мұқият түрде жоспарланған жағдайда, сервердегі қолданушының ақпараттың алу ықтималдылығы арта түседі. Нәтижесінде зиянкес қолданушының атынан отырып, ауқымды зардап тигізу мүмкін.

Қауіпті кодты еңгізу шабуыл түрі үшке бөлуге болады:

– тұрақсыз;
– тұрақты сақталған;
– құжаттардың объекті үлгісі.

Тұрақты сақталған және тұрақсыз шабуыл арасында негізгі айырмашылығы, серверге жіберілген қауіпті код пайдаланушыға кайту кезеңі бір HTTP сұрау арасында өтеді. Тұрақты сақталған түрінде сұраулар әртүрлі болады.

Тұрақсыз шабуыл түрін іске асыру үшін, пайдаланушыны зиянкестің жасалған сілтемесіне көшугі талап етеді. Бұл дегеніміз қолданушының шолғышындағы URL-дің ішіне зиянкестің сілтеме мекен-жайының жазылуы арқылы орындалады.

Сақталған шабуылдың орындалу жағдайлары негізіне қолданушының почталары және танымал форумдарда немесе әртүрлі қорғалмаған веб-сайттарда орындалады. Әдетте бұл шабуылды жүзеге асыру үшін қолданушы сілтеме арқылы көшу міндетті емес, қорғалмаған мекен-жайға өтудің өзі жеткілікті. Мысалы: көптеген сайттарда қолданушы ақпаратты қалдыру мүмкіншілігі бар форумдарды қолданады. Ақпаратты қалдыру үшін қолданушы міндетті түрде тіркелу қажет. Тіркелу кезеңі барысында қолданушының арнайы нөмірі сервердің деректер қорына енгізіледі және cookie құжатында сақталады. Шабуылдаушы қолданушының шолғышында

Javascript бағдарламалау тілінде жасанды хабарды жіберу арқылы сақталған құжатты қатынайды.

Шабуылдаушы осал веб-қосымшалардың ғаламтор аралық скриптинг арқылы тегтер енгізген кезден бастап осалдық аймағында жатады. Бұл шабуыл түрі әзірлеудің екі жағында жұмыс істей алады. Бірақ қолданушы жағындағы әзірлеу кезеңінде жиі кездеседі. Javascript бағдарламалау тілі арқылы енгізілген кодты былай қолдануға болады:

– веб-қосымшаның куки құжаттына рұқсат алу;
– веб-қосымшаның сыртқы түріне өзгерістер енгізу;
– буферді тікелей қатынау немесе қолдану мүмкіншілігі;
– қолданушының пернелерге енгізген ақпаратты бақылау.

Құжаттардың объекті үлгісі арқылы іске асатын шабуыл түрі қолданушының шолғышында құжаттың үлгісін өзгерту арқылы жұмыс істейді. Көп жағдайда бұл шабуыл бағдарламалау тіліндегі тәсілді қолдану арқылы құжаттың объект үлгісін innerHTML-мен өзгертеді. Жазылу реті келесі түрде кездеседі:

nurzhan.innerHTML = `<link href="./nurzhan/${nurzhanId}/nurzhan.css">`;

Жоғарыда көрсетілгендей, бағдарламалау тілінің тәсілін қолдану арқасында шабуылдаушы қолданушыны жалған сілтемеге апарады. Қолданушы сілтемеге кіркен сәттен бастап қолданушының ақпараттары арнайы құжатта сақталады. Зиянкес бұл құжаттарды өзінің қашықтағы серверіне жіберіп, қолданушының ақпаратың өз мақсаттарында қолдануды жүзеге асырады.

2.2 Вирустік шабуылдар

Қолданушылардың көпшілігі қазіргі таңда ғаламторды қолданып, керекті ақпаратты немесе қызметті ала алады. Алайда ғаламтор ішінде зиянкестердің арқасында вирус жұқтыру қауіпі арттады. Вирусты жұқтыру алгоритмі қолданушыларды әртүрлі күдікті ресурстарға апару және сілтемелерге көшіру арқылы жүзеге асырады. Ғаламторда вирустік шабуылдың кең тараған түрлеріне:

– баннер арқылы қолданушыны қауіпті сілтемеге көшіру. Баннерлер желі ішінде қолданушыға қызығушылық танытатын кейіпте кездеседі. Қолданушының шолғышында қауіпті сілтемеге көшу алдында сұрау қызметі орнатылмаған жағдайда баннер арқылы қауіпті вирусті жұқтыру қауіпі артады;

– қалқымалы терезелер. Әдетте осындай терезелер қолданушының оң жақ бұрышында жиі орналасады. Пайдаланушы мұндай терезеге басқан сәттен вирустік бағдарлама жүктеледі. Жүктелу үдерісі қолданушының шолғышындағы кеңейту құжаттамасында сақталады. Сол себебтен ғаламторға кірген сәттен бастап осы вирус іске қосылады. Көптеген қолданушылар қалқымалы терезелерді жабу үшін жабу батырмасын басады. Алайда ол батырманың ішінде қауіпті сілтеме орналасады. Осының арқасында қолданушы тағы бір вирусті жүктейді;

– спам тарату. Қолданушыға хабарламаларды рұқсатсыз жіберу арқылы іске асырылатын вирустік шабуыл - спам деп аталады. Бұл шабуыл ақпаратты таратуға қол жетімді ақпарат көздерінде колданады, бірақ көбінесе электрондық пошта арқылы іске асырылады.

Спам таратудың бірнеше критерийлері болады. Таралу аймағына сәйкес спам тарату ғаламторда немесе офлайн іске асырылады. Ғаламторда тарату әдісі хабарламаларды автоматтандыру бағдарламасының арқасында жіберіледі. Офлайн таралу үдерісінде спам қолмен таратылады.

Спам тарату арқылы қолданушының жеке деректерін алуға, компьютеріне вирус жұқтыруға, онлайн әмиянға кіру және басқа қауіпті істерді жүзеге асырады. Шабуылдаушы хаттама құрамына қауіпті сілтемелер немесе бағдарламаны енгізеді. Мысалы хаттама ішінде зиянкес өзін банк қызметкері ретінде таныстырып, қауіпті сілтеме арқылы көшуңізді сұрайды. Сілтеме арқылы қолданушы жеке ақпараттарын, құпия сөздерін толтырып, зиянкестің серверіне жібереді. Хаттама ішінде бағдарлама немесе зиянды құжаттар болу қауіпі бар. Осындай құжаттарды жіктеу арқасында қолданушының жеке деректерің құпия түрде зиянкестің бағдарламасына жиналады.

2.3 Басқару жүйелерін бұзу

Қазіргі таңда кез келген веб-қосымшаларды әзірлеудің негізгі екі түрі кең тараған. Бірінші түрі әзірлеу құралдары арқылы күрделі қосымшаларды жасау, ал екінші түрі дайын веб парақшаларды орналастыру арқылы құрылатын басқару жүйелері негізінде. Басқару жүйелері веб-қосымшаны жиі өзгерту немесе жаңарту керек болған жағдайда қолданады. Өйткені басқару жүйелері динамикалық түрде қосымшаның мазмұның өзгертеді. Ал әзірлеу құралдары статикалық түрде жұмыс істейді. Басқару жүйесі әзірлеу құралдарынан басты артықшылығы ол, веб-қосымшаның құрылымын барлық парақшаларда тұтас өзгертеді.

Басқару жүйелері басты міндеті қосымшаға түсетін ақпаратты веб сервердің көмегімен бақылау арқылы қамтамасыз ету. Бұл жүйені қолдану барысында құжаттарды іздеуге, қайталанатын және қателіктерді алдын алуға, қолданушылармен байланыс уақыттың қысқартуға болады.

Басқару жүйелерін күрделі үш топқа бөлуге болады:

– бірінші топқа әзірлеу мамандары жасайтын қосымшалар жатады. Бұл қосымшалар басқару жүйелерінің мүмкіндіктерін кеңейту мақсатында әзірленеді;

– екінші топ ішіне әкімші қосымшаның құрылымын өзгертетін басқару жүйелері;

– үшінші топ құрамына қолданушының іс-әрекеттерін бақылауға арналған басқару жүйелері.

Дегенмен, басқару жүйесін веб-қосымшаларда қолдану барысында кемшіліктер туындайды. Қолданушы басқару жүйелері арқылы сұраулар жіберу үдерісі қарапайым веб-қосымшаға қарағанда баяу жұмыс істейді. Егер

сұраулар саны көп болған жағдайда басқару жүйесі істен шығу мүмкіншілігі туындайды. Тағы бір кемшілігі, басқару жүйесі бір платформаға тәуелді болады.

Басқару жүйелерін бұзу үдерісін әртүрлі тәсілдер арқылы жүзеге асыруға болады. Ең көп тараған тәсілге басқару жүйесінің қашықтық серверге орнатылған нұсқасын білу арқасында бұзу жатады. Егер қолданушы басқару жүйесін жаңа нұсқасын жүктемеген жағдайда осы тәсілді қолдануға болады. Бұл тәсіл арқылы зиянкес басқару жүйесінің нұсқасын біліп, сол нұсқадағы осал тесіктері арқылы қолданушының маңызды ақпараттарын деректер қорынан немесе қолданушының шолғышында сақталған ақпарат арқылы ұрлай алады.

Егер басқару жүйелерін ресми жүктеу көздерінен алынбаған жағдайда, пайдаланушыларға зиян тигізу ықтималдылығы артады. Зиянкес жалған басқару жүйелері ішіне өзіндік қосымша модульді енгізеді. Бұл модуль арқылы барлық қолданушылардың жеке ақпараттарын жеке құжатта сақтап, зиянкестің қолына түседі.

Ескі басқару жүйелерінде кірістірілген осалдықтар талдағыштары болмағандықтан, зиянкес қолданушының атынан қауіпті сұрауларды жібере алу мүмкіншілігі пайда болады. Кейбір қолданушылар басқару жүйесін функцияларын кеңейту мақсатында қосымша плагиндерді орнатады. Ол қосымша плагиндер басқару жүйелердің нұсқаларымен сәйкес болмаған жағдайда істен шығу қауіптері пайда болады.

2.4 Веб-қосымшалардың қауіпсіздік шаралары

Хакерлер ғаламторда жалпыға қолжетімді веб-қосымшаларды бұзу арқылы табыс көздеріне жету үдерісі қызықтырады. Google сарапшылары 2019 жылы хакерлік шабуылдар саны 2018 жылмен салыстырғанда 38% - ға артқаның хабарлайды [5]. Сондықтан веб-қосымшалар үшін қауіпсіздік шараларын ұстанған абзал. Төменде сипатталған қауіпсіздік шараларын қолдану арқылы қорғау мәселелерін шешуге көмектеседі.

Веб-қосымшада HTTPS хаттамасын қолдануға болады. Қолданушының маңызды ақпараттарын Hyper Text Transfer Protocol Secure арқылы шифрлап, ғаламтор арасында таралу кезінде ақпаратты қорғайды. HTTPS ақпараттың тұтастығын және құпиялығын қашықтағы сервермен әрекеттесу арқасында сенімді қорғаныста қамтамасыздандырады. Қолданушы веб-қосымшада өзінің каржы туралы ақпаратты, нөмір картасын немесе жеке деректерді еңгізген жағдайда HTTPS-ті қолданған дұрыс. Веб-қосымшаның барлық беттерінде осы хаттаманы қолданбаған жағдайда, шабуылдаушы қолданушы толтырған ақпаратты cookies құжатта сақтап, қолданушының атынан серверге жалған сұраулар жібереді. Осындай жағдайға тап болмау үшін HTTPS хаттамасын веб-қосымшаның барлық парақшаларында қолданған дұрыс шешім болып табылады. Бұл хаттаманы SSL сертификатпен серверге қосу арқылы қауіпсіздікті жоғары деңгейге көтеруге болады. Қазіргі уақытта HTTPS

хаттамасы орнатылған веб-қосымшалар сенімді болғандықтан, ғаламторда іздеу нәтижесінде алдынғы қатарда орналасады. Яғни ескі HTTP хаттамасы іздеу барысында HTTPS хаттамасымен салыстырғанда соңғы қатарларды құрайды.

Бағдарламалық қамтамасыз ету жүйелерін жаңартып отырыңыз. Шабуылдаушы веб-қосымша қолданған басқару жүйелерін және бағдарламалық қамтамасыз ету жүйелерін осалдықтарын үнемі бақылайды және қолдануға дайын тұрады. Сол себебтен қолданушы үнемі жаңа нұсқадағы бағдарламалық қамтамасыз ету жүйелерін жаңартып отыру қажет. Әдетте барлық веб-қосымшалар ғаламтор арқылы жұмыс істегендіктен, жаңа нұсқалары ғаламторды қосқан сәттен қолданушының құрылғысына жазылады. Бірақ пайдаланушы бұл бағдарламалық қамтамасыз ету жүйелерін жаңартудан бастартуы мүмкін. Оның себебтері әртүрлі болады, ең кең тарағандары:

– қолданушы ғаламтор желісінің нашар болуы;
– қолданушының жадысы толуы;
– жаңа нұсқаларды қолдау мүмкінділігі болмауы;
– қате жүктеу.

Ақпаратты сұрыптау қосымшалары. Сайт аралық скриптинг веб-қосымшада болдырмау мәселесін қарастыратын болсақ, бұл шабуыл қолданушының құрылғысында жұмыс істейтін, веб-қосымшаның құрылымын өзгерту арқылы ұрланған деректерді зиянкеске жіберу түрінде жұмыс істейді. Осы шабуыл түрі көп жағдайда авторизация парақшаларында қолданылады. Қолданушы авторизация формасында толтырған ақпаратты cookie құжатында сақтайды, зиянкес ол ақпаратты серверге сұрау жіберу кезінде cookie құжатты өзіне жүктейді.

Сайт аралық скриптинг болдырмау үшін, тексеру барысында шолғышқа толтырылған ақпаратты сұрыптау қажет. Ол үшін динамикалық жасалған қызметтің мәндерін өзгертіп, таңбалардың шаблондық жүйелерін қолданады. Тағы бір қауіпсіздік шарасына кіретін тәсіл, басқа домендегі скриптердің орындалуына және қызметтерді өшіруге тыйым салу арқасында қауіпсіздікті орнату.

Құпия мәліметтерді шифрлау және тексеру. Веб-қосымшаға құпия ақпаратты деректер қорында хэш түрінде сақтаған қауіпсіздік шараларының бір түрі ретінде саналады. Осындай қауіпсіздік шарасы арқылы қолданушының құпия сөздері деректер қорындағы хэш түрінде сақталған деректермен салыстырылады. Деректер сәйкес болған жағдайда ғана веб-қосымша қолданушыны бағдарламаға кіргізеді. Егер қаскүнем қосымшаны бұзып құпия ақпаратты және хэш түрінде сақталған деректерді алған жағдайда, хэшті қайтадан бастапқы түрге келтіру мүмкін емес. Бірақ хэш түрінде сақталған ақпарат қарапайым немесе танымал парольден тұратын болса, бұзу ықтималдылығы көбейеді. Шабуылдаушы хэшті бұзу үдерісі баяу және көп ресурстардан бас тартуға мәжбүр етеді. Нәтижесінде үлкен шығындарға әкеледі.

Құпия сөзді енгізу барысында құпия сөздің ұзындығына шектеу қойып, қолданушының логині, тіркелген электрондық пошта немесе жеке нөмірімен сәйкестігін тексеру керек. Қазіргі таңда көптеген басқару жүйелерінде, бағдарламалық жүйелерде парольдік кауіпсіздік орнатылған. Егер бұл қауіпсіздік шарасы болмаған жағдайда дайын қосымша модульдерді орнату арқылы қауіпсіздікті қамтамасыз етуге болады.

Құпия сөзді енгізу барысында сандарды, әріптерді және арнайы таңбаларды қолданған дұрыс. Таңбалардан (“$”, “@”, “&” және басқалар), бас және кіші әріптермен араластырып және сандарды қосу арқылы сенімді құпия сөзді құрастыруға болады. Мысалы, қарапайым nurzhan123-тің орнына, Nu1zh@N_1^3 құпия сөзінің сенімділігі арта түседі. Бірақ бір құпия сөзді барлық жүйелерде қолданған дұрыс емес. Сондықтан әр жүйе, бағдарлама, жеке аккаунт және басқада платформаларда әр түрлі құпия сөзді қолданған жөн.

Міндетті түрде осындай құпия сөздерді басқа құрылғыларда, жеке сервистерде, аккаунттарда пайдаланған қауіпсіздікті қамтамасыздандырады.

2.5 Сканерлеу

Көптеген компаниялар қолданушылар саның арттыру үшін ғаламтор арқылы веб-қосымшаларды енгізеді. Қолданушылармен ақшалай айырбастау, қашықтан қызметтер алу және басқа мүмкіншіліктер пайда болуына байланысты, веб-қосымшалардың танымалдығы кеңейе түсті. Алайда веб-қосымшалар ғаламторда орналасуына байланысты, қолданушылардан басқа шабуылдаушыларда көре алады. Осыған байланысты шабуылдаушы веб-қосымшаның осал жерлеріне назар аударады. Сол себебті әзірлеуші шабуылдарға төтеп беру үшін веб-қосымшаның әзірлеу кезеңінен кейін осал жерлерін тексеру қажет. Веб-қосымшалардың қауіпсіздік деңгейін арттыру үшін әртүрлі құралдар жиынтығы қолданылады. Қауіпсіздік деңгейін қамтамасыздандырудың ең тиімді құралы осалдықтарды іздеу болып табылады, яғни веб-қосымшаларды сканерлеу. Сканерлеу - дегеніміз веб-қосымшада кездесетін ақауларды іздеу арқылы қолданушылардың жеке деректерін сақтауға және ұрлаудан сақтайтын бағдарламалық-аппараттық құрал. Сканерлеудің арқасында веб-қосымшада жиі кездесетін осалдықтар жиынтығын анықтауға болады. Олар келесі санаттарға бөлінеді:

– қауіпті бағдарламалық кодтың осалдығы;
– веб-қосымшаны конфигурациялау осалдығы;
– енгізу осалдығы.

Қауіпті бағдарламалық кодтау осалдылықтарына ақпараттың жіберуі және жауабтың дұрыс өңдеу үдерісінде жүргізіледі, яғни веб-сервеге сұрау жіберу және серверден сұрау жаубын күту арасында ұйымдастырылады. Бұл осалдықтар қатарына жиі кездесетін ғаламтор аралық скриптинг XSS және SQL-инъекция жатады.

Конфигурациялау осалдықтарына веб-қосымшаның ішінде жүргізілетін кезендердің қате баптауларына байланысты туындайтын осалдықтар жатады. Мысалы: SSL/TLS хаттамаларының қате конфигурациялау, фреймворкті немесе қосымша кітапханаларды серверге дұрыс енгізбеу, басқа компоненттердің сәйкеспеуі және басқалар.

Енгізу осалдылықтары негіздеріне бағдарламалық қамтамасыз ету жүйелерінің ескіруі, қарапайым парольдерді пайдаланумен, қызметтік модульдердің қолжетімді веб-серверде мұрағаттық көшірмелерді сақтау нәтижесінде туындайды.

Сканерлеу үдерісі барысында веб-қосымшалардың жалпы жағдайда жұмыс істеу принципі осылардан құралады:

– веб-қосымшаның осал жерлері туралы деректер жинау;

– аудит жүргізу негізінде веб-қосымша аққаулықтарын осалдықтар қорынан анықтау:

– бағдарламалық жүйенің әлсіз жерлерін табу;

– табылған осалдықтарды жою туралы ұсыныстар беру.

Веб-қосымшалардың қорғау мақсаттарына байланысты сканерлерді келесі түрлерге бөлуге болады:

Желілік сканер арқылы қол жетімді желілердің нұсқаларын біліп, серверде операциялық жүйелердің орнатылуын анықтауға мүмкіншіліктер береді.

Веб-қосымшада бағдарламалау тілдері арқасында орындалатын қауіпті скриптерді анықтайтын сканерге веб скриптердің осалдық сканері жатады. Бұл сканердің көмегімен бағдарламалық қамтамасыз ету және автоматты түрде орындалатын бағдарламалық кодты іздеуге негізделеді.

Эксплойттар веб-қосымшаның ішінде жасырын түрде жұмыс істеп, ақпаратты жеке құжатта сақтайды. Сол себебті эксплойттарды іздеуге арналған сканерлерді веб-қосымшада қолданған дұрыс. Сканер инъекцияларды және эксплойттарды веб-қосымшаның барлық жүйелік файлдарында автоматты түрде іздейді. Қосымша қателер пайда болған жағдайда, веб-қосымшаның толық түрде функцияналды жұмысын тиімді жолмен шешуге болатын жолдары ұсынылады.

Жұмыс барысында веб-қосымшаны толық түрде сканерлеу 5 кезеңнен тұрады:

1) Ақпаратты жинау үдерісі. Ақпарат жинау үдерісі барысында қосымшада қолданылған барлық құрылғыларын сарапталып, құрылғыларға қосылған қызметтер анықталады. Операциялық жүйенің қауіпсіздігін талдау кезеңі автоматты түрде келесі кадамға өткізіледі, өйткені сканерлеу әрбір кезектің түйінде жүйелік сканерлер орнатылады.

2) Осалдықтың ықтималдылығын анықтау. Бұл кезеңде сканер деректер қорынан осалдықтың тақырыптарын және белсенділігін тексеру арқасында табылған осалдылықтармен салыстырылады. Сканерлеу барысында табылған осалдықтан қауіп-қатер бойынша ретпен тізіледі. Көптеген сканерлерде

осалдықтар қауіпі үш дәрежеге ие болады: жоғары қауіп дәрежесі, орташа қауіп дәрежесі және төменгі қауіп дәрежесі.

3) Анықталған осалдықтарды тексеру. Осалдықты тексеруге әр сканер арнайы әдістер мен белгілі бір тәсілдерді қолданып, тыбылған осалдықтардың қауіп-қатер дәрежесіне бөледі. Осы дәрежелерге байланысты осалдықтың қауіп-қатері бойынша жою туралы шешім қабылданады.

4) Осалдық туралы есеп беру үдерісі. Қауіпсіздікті талдау жүйесі жиналған ақпарат бойынша анықталған осалдықтарды сипаттайтын есептер нәтижесін шығарады. Есеп шығару барысында қолданушылардың санаттарына карай қосымшада қарапайым қолданушыдан бастап желілік администраторларға дейінгі деректер есеп нәтижесі ретінде шығарылады. Есеп шығару нәтижесінде осалдық туралы ақпарат графиктер және кестелер түрінде қолданған нақты жасалған есептерді ұсыну қажет және анықталған осалдықтарды жою бойынша тиімді ұсыныстарды көрсетуге болатын қызмет болуы қажет. Әрбір табылған осалдық үшін ақауды жою туралы үдерісі қадамдар арқылы жүргізілетін нұсқауларға сүйену негізінде жұмыс істейді.

5) Табылған осалдықтарды жою. Осалдықтарды немесе ақауларды жою қызметі барлық сканерлерде орнатылмаған. Қосымшаға зиян осалдықтарды жою қызметін сканерге орнату үшін, ең алдымен сканердің ресми көздерінен қосымша модульдерді жүктеу керек. Жүктеу барысында сканер және жүктелген құжаттардың нұсқалары сәйкес болуы тиіс. Егер сәйкестік болмаған жағдайда веб-қосымшаны сканерлеу барысында осалдықтардың жиналған деректері жалған түрде көрсетіледі. Көп жағдайда желілік сканерлерде осалдықты жою қызметі орнатылмайды, алайда осындай қызмет жүйелік сканерлердің барлық нұсқаларында қолданылады.

Әзірлеуші немесе желілік администратор веб-қосымшаның сканерлеу үдерісінің мүмкіндіктерің әртүрлі амалдармен кеңейтуге рұқсат береді. Сканердің мүмкіндіктерін кеңейту үдерісі бағдарламалау тілдері арқылы сканердің жүйелік файлдарға скрипт енгізумен арқылы жүзеге асырылады. Скрипт арқылы веб-қосымшада табылған осалдықтарды жою үдерісі барысында басқару жүйелерін сканерлеу тарихын сақтамау, сканерлеу барысында веб-қосымшаға тигізетін қауіп-қатер дәрежесіне қарай сұрыптау үдерісін автоматтандыруға және сканерлеудің уақыт бойынша тексеру аралығын орналастыру мүмкіндіктерің енгізуге болады. Алайда сканерлеу мүмкіндіктерін кеңейту барысында сканер толық веб-қосымшаның осалдықтарын тексереді. Сол себебті енгізілген скриптің орындалуына байланысты жүйелік өзгерістерді болдырмайтын косымша бағдарлама әзірленеді. Әзірленген қосымша бағдарлама арқылы веб-қосымшаның қалыпты жұмысы бұзылған жағдайда автоматты түрде іске қосылады. Яғни осалдықты жою барысында жүйелік бұзылған құжаттардың жағдайын қайта келтіру жүргізіледі. Сондықтан сканерді кеңейту нәтижесінде веб-қосымшаның осалдықтарын жою кезеңі сапалы өткізіледі.

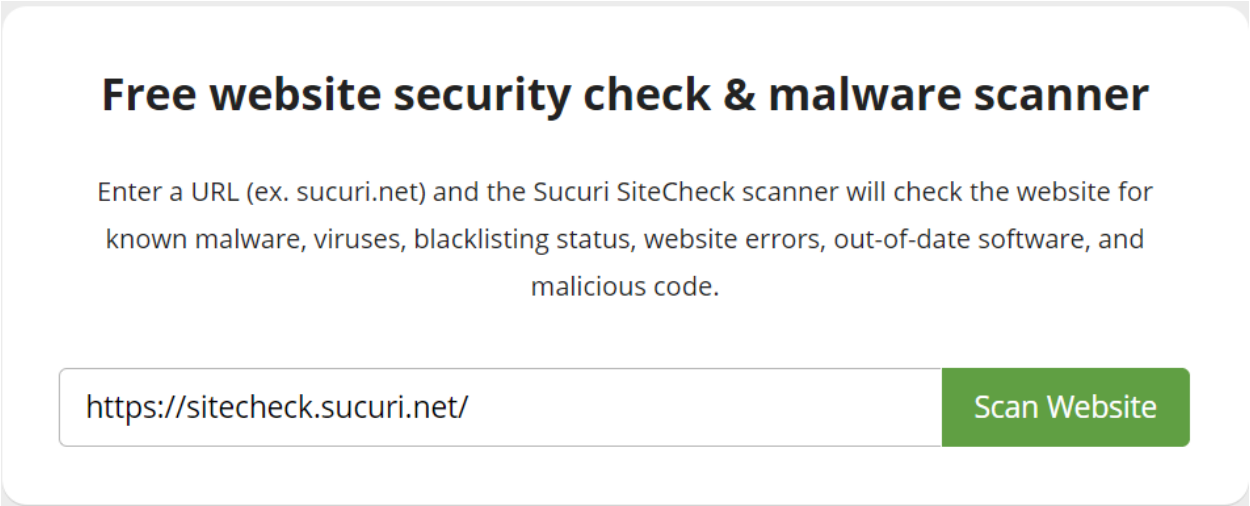
Осалдықтарды іздеу жағдайында, әкімшінің қорғану тәсілдері және қауіпсіздікті талдау жүйесін тиімді қолданудың бірнеше әдістері бар:

– ықтимал осалдылықтарды сканерлеу арқылы тексеріп іске асыру. Басқа әдістермен салыстырғанда бұл тексеру осалдықты нақтылай анықтап және сонымен қатар табу жылдамдағы арттырады.

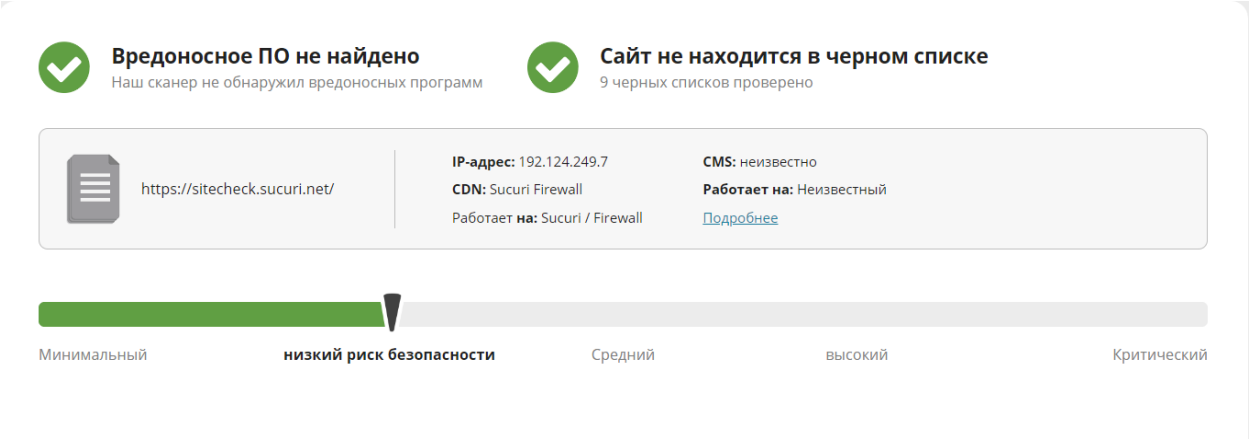
– сканерлеу нәтижесінде расталған осалдықтарды тексерумен сканерлеуді орындау. Осы әдіс барысында расталған осалдықтарға сараптама жүргізіліп, жүйелік хосттың орындалу кезеңдеріне зардап тигізу қауіпі болады.

– осалдықты табу негізінде қолданушының ережелерін толық түрде сканерлеу әдісін жүзеге асыру.

Sitecheck.sucuri.net сканерлеу сервисі арқылы осалдықтарды табуға болады:

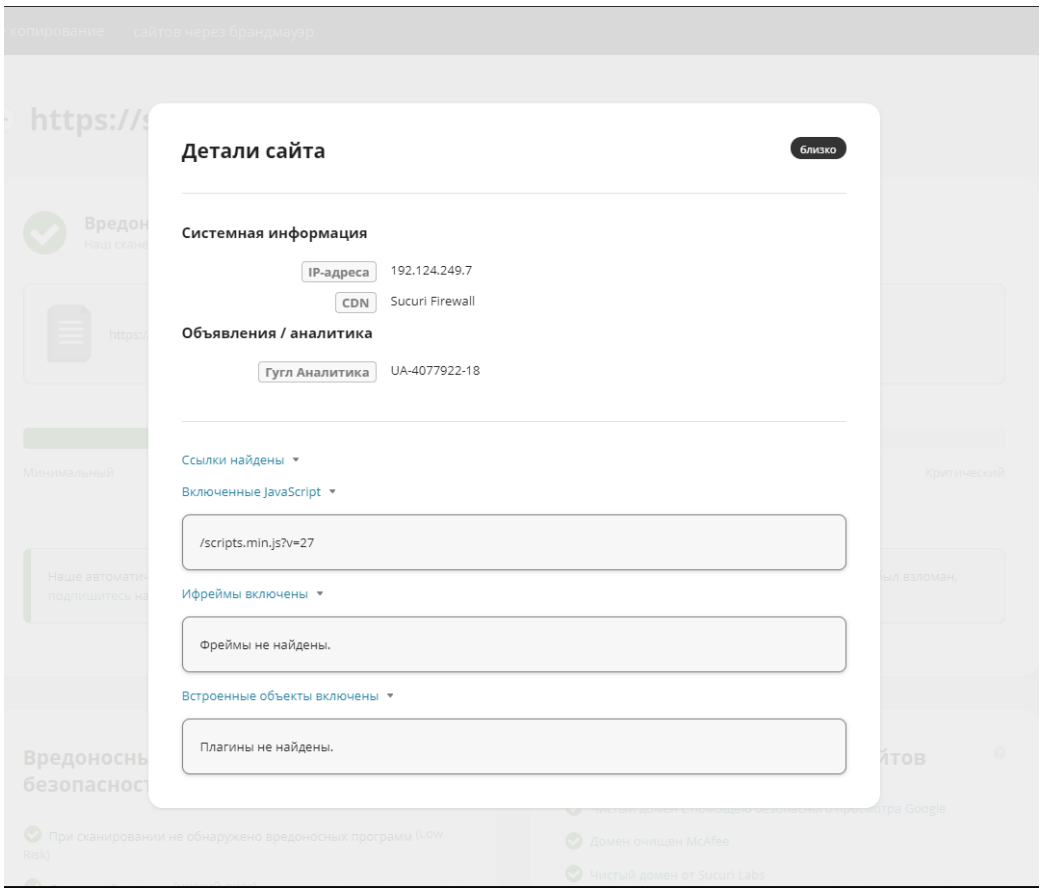


2.1 сурет – Сканерлеу үдерісі



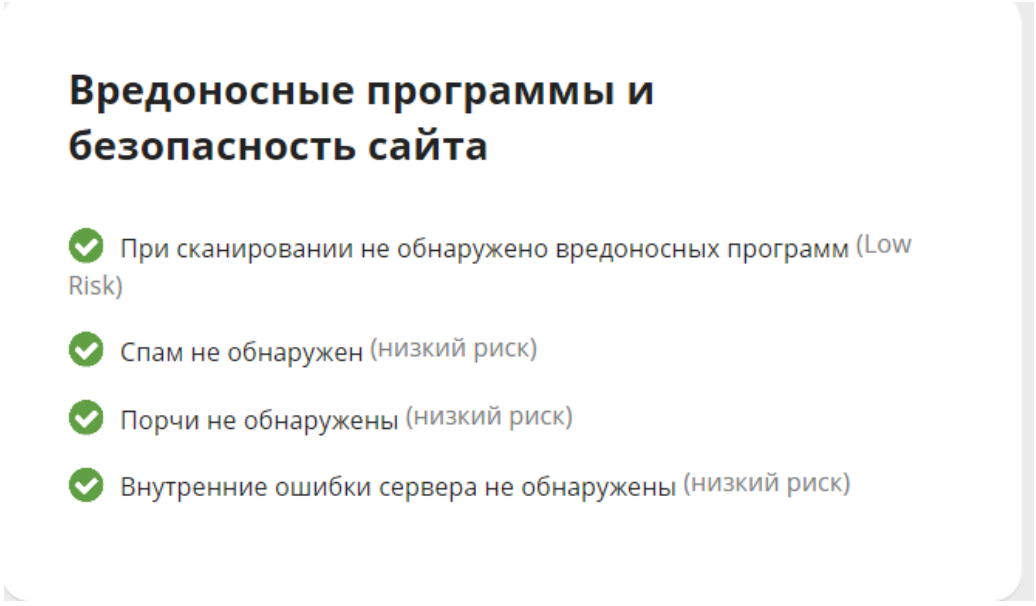
2.2 сурет – Сканерлеу нәтижесі

Сканерлеу барысында веб-сайттың жүйелік деректері және аққау келтіретін плагиндер анықталады. Сканер Javascript бағдарламалау тілі арқылы жасалатын ғаламтор аралық скриптинг шабуылдарын тексеруге мүмкіндіктер береді. Көптеген веб-қосымшаларда қолданушылардың кандай платформадан, қолданушының орналасқан мекенің анықтауға және қолданушы қандай іс-әрекет жасаганың аналитикалық бағдарлама графиктер немесе құжаттар ретінде серверде сақталады. Сканер веб-қосымшада қосылған аналикалық құралдардың осалдықтарына байланысты тексереді.



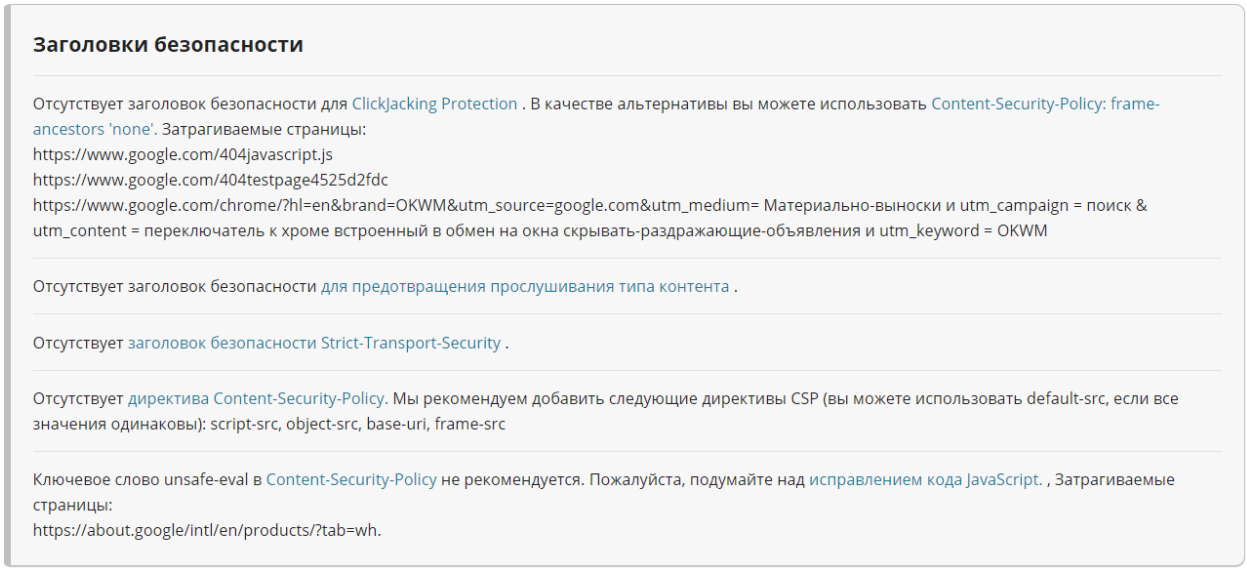
2.3 сурет – Сканерлеу бойынша веб-сайттың деректері

Қосымшаны тексеру үдерісі бірнеше қадамдардан тұрады. Сканерлеу ең алдымен серверде және қолданушы жағындағы осалдықты тексеру қажет. Тексеру үдерісі табылған осалдықты дерек қорындағы осалдықтармен салыстырылып сараптау кезеңіне көшіріледі. Осалдықтың қауіп-қатеріне байланысты дәрежелерге сұрыптау жүргізіледі. Осалдықтың дәрежесіне сәйкес жою туралы ұсыныстар қабылданады.



2.4 сурет – Тексеру қадамдары

Ең соңғы кезеңде сканер тексеру барысындағы осалдықтар тізімі көрсетіледі. Табылған осалдықтарды тиімді жою туралы шешімдер ұсынылады.

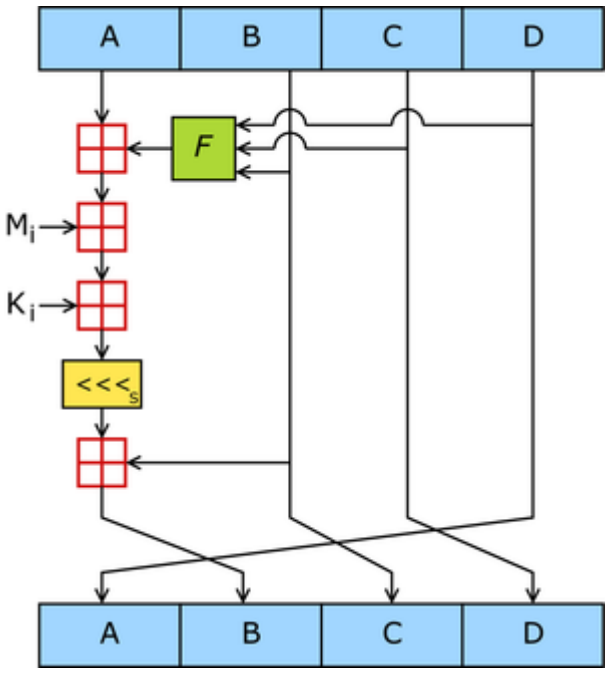


2.5 сурет – Табылған осалдықтарды жою ұсыныстары

Осындай кезеңдерден тексерілген веб-қосымшалар кауіпсіздік деңгейін жоғары сапада көтеруге мүмкіндік береді. Әр кезеңнің өзіндік модульдердің қосылуымен тексеру үдерісін сапалы түрде жүзеге асырылуын қамтамасыз етеді. Модульдерде қауіпті жою туралы тиімді немесе жаңа жою тәсілдері жүктелінеді. Толықтай тексерілген веб-қосымша ғаламтор беткейінде іздеу бірінші парақшаларда жайғасады. Осыған байланысты қолданушылардың сенімін ақтау арқылы веб-қосымшаның танымал болуына септігін тигізеді.

2.6 MD5 алгоритмі

MD5 (Message Digest 5) — хабарлама дайджесттерін құруға арналған алгоритмдер қатарының біріне кіреді. Бұл алгоритм еркін ұзындықтағы хабарлама дайджестерін жасап, 128 биттік хеширлеу арқылы жүзеге асырылады. MD5 көп жағдайда құпия сөздердің хештерін сақтауға және ақпараттың бүтіндігін тексеруге үшін веб-қосымшаларда қолданылады. Бастапқы мәнге алгоритм хабарламаларды, құпия сөздерді немесе кез келген мәтіндерді қабылдап, нәтиже ретінде хешталған деректерді шығарады. Алгоритм хешталған деректерді кері шифрлауға мүмкіндік бермейді және шифрланған деректердің түпнұсқалығын тексеру үшін қолданылады. Сол себебті алгоритм барынша қауіпсіздікті қамтамасыздандырады.



2.6 сурет – MD5 алгоритмінің сұлбасы

Алгоримт кіріс деректер ағының қабылдау арқылы ақпараттың хештеріне сәйкес ізделеді. Үдеріске кірген деректердің ұзындығы бит ретінде өлшенеді. Енгізілген деректердің ұзындығы бүтін санды қабылдап және теріс мәнді қабылдамау қажет. Деректер толықтай түскеннен кейін, ағындарды есептеу үдерісіне дайындау жүргізіледі.

Алгоритм 4 қадамдан тұрады:

1) Ағынды түзету қадамы. Бұл қадамда ағынның соңғы жеріне бір бит жазылады. Соңғы бит жазылғаннан кейін бір нөлдік бит қосылады. Нәтижесінде ағынның ұзындығы 448-ден 512 модульге дейін салыстырыла алатын мүмкіндік береді. Ағынның түзету қадамы кез келген сәтте жүргізіледі, тіпті ағынның бастапқы ұзындығы 448-ге тең болған жағдайларда жүзеге іске асырылады.

2) Деректердің ұзындығын қосу. Деректердің соңына 64-биттік ұзындықты құрайтын қосымша деректер тураланудың алдында енгізіледі. Бастапқыда 4 кіші байт жазылады және келесі жоғарғы байттар жазылады. Деректің ұзындығы $2^{64} – 1$ – ден асатын жағадай болса кіші биттер жазылып, ағынның ұзындығы 512-ге көбейтіледі. Хабарламаны есептеу барысында массив ретінде қалыптасқан 512 биттік ағынға негізделеді.

3) Буферді инициализациялау қадамы. Деректерді есептеу үшін 32 бит көлемінде 4 айнымалы іске қосылады және little-endian байт форматы ретінде бастапқы мәндерді он алтылық сандары беріледі:

A = 01 23 45 67; // 67452301h;

B = 89 AB CD EF; // EFCDAB89h;

C = FE DC BA 98; // 98BADCFEh;

D = 76 54 32 10. // 10325476h.

Жоғарыда көрсетілген айнымалылардың ішінде сақталған аралық есептеулер нәтижелерін құрайды.

1-і раунд: $\mathrm{FunF}(X, Y, Z) = (X \wedge Y) \vee (\neg X \wedge Z)$,

2-і раунд: $\mathrm{FunG}(X, Y, Z) = (X \wedge Z) \vee (\neg Z \wedge Y)$,

3-і раунд: $\mathrm{FunH}(X, Y, Z) = X \oplus Y \oplus Z$,

4-і раунд: $\mathrm{FunI}(X, Y, Z) = Y \oplus (\neg Z \vee X)$,

2.7 сурет – Төрт раундтің цикл айнымалысы

Есептеу үшін төрт раундтан құралған төрт функция қолдану қажет. Функция үш параметрді қабылдау арқылы нәжисінде сөз шығарылады. 4 кезеңнің 16 раундтық есептеулерден әрбір 512-биттік блоктар өткізіледі. Блок массив ретінде 32 биттік 16 сөзден құрылада.

4) Цикл арқылы есептеу қадамы. Бұл қадам бойынша есептеу n элементі деректер блогын 512-биттік блоктар массивтерінен алынады. Бастапқы блоктарда жүзеге асырылғаннан кейін келесі A, B, C, және D шығарылған бастапқы мәндерін сақтайды. Мысалы:

– AA = A;
– BB = B;
– CC = C;
– DD = D.

```
/* [abcd k s i] a = b + ((a + F(b,c,d) + X[k] + T[i]) <<< s). */
[ABCD  0 7  1][DABC  1 12  2][CDAB  2 17  3][BCDA  3 22  4]
[ABCD  4 7  5][DABC  5 12  6][CDAB  6 17  7][BCDA  7 22  8]
[ABCD  8 7  9][DABC  9 12 10][CDAB 10 17 11][BCDA 11 22 12]
[ABCD 12 7 13][DABC 13 12 14][CDAB 14 17 15][BCDA 15 22 16]
```

2.8 сурет – Бірінші циклдің кезеңі

```
/* [abcd k s i] a = b + ((a + G(b,c,d) + X[k] + T[i]) <<< s). */
[ABCD  1 5 17][DABC  6 9 18][CDAB 11 14 19][BCDA  0 20 20]
[ABCD  5 5 21][DABC 10 9 22][CDAB 15 14 23][BCDA  4 20 24]
[ABCD  9 5 25][DABC 14 9 26][CDAB  3 14 27][BCDA  8 20 28]
[ABCD 13 5 29][DABC  2 9 30][CDAB  7 14 31][BCDA 12 20 32]
```

2.9 сурет – Екінші циклдік кезеңі

```
/* [abcd k s i] a = b + ((a + H(b,c,d) + X[k] + T[i]) <<< s). */
[ABCD  5 4 33][DABC  8 11 34][CDAB 11 16 35][BCDA 14 23 36]
[ABCD  1 4 37][DABC  4 11 38][CDAB  7 16 39][BCDA 10 23 40]
[ABCD 13 4 41][DABC  0 11 42][CDAB  3 16 43][BCDA  6 23 44]
[ABCD  9 4 45][DABC 12 11 46][CDAB 15 16 47][BCDA  2 23 48]
```

2.10 сурет – Үшінші циклдік кезеңі

```
/* [abcd k s i] a = b + ((a + I(b,c,d) + X[k] + T[i]) <<< s). */
[ABCD  0 6 49][DABC  7 10 50][CDAB 14 15 51][BCDA  5 21 52]
[ABCD 12 6 53][DABC  3 10 54][CDAB 10 15 55][BCDA  1 21 56]
[ABCD  8 6 57][DABC 15 10 58][CDAB  6 15 59][BCDA 13 21 60]
[ABCD  4 6 61][DABC 11 10 62][CDAB  2 15 63][BCDA  9 21 64]
```

2.11 сурет – Төртінші циклдік кезеңі

Жоғарыда көрсетілген циклдардың нәтижесін шығарсақ:

A = AA + A

B = BB + B

C = CC + C

D = DD + D

Циклдің есептеу үдерісі толықтай аяқталған кейін, тексеру барысында циклдан өтпеген блоктарды табылған жағдайда цикл қайта орындалады. Қайта орындалу кезінде массивке n+1 элементі өтіп циклді кайталайды.

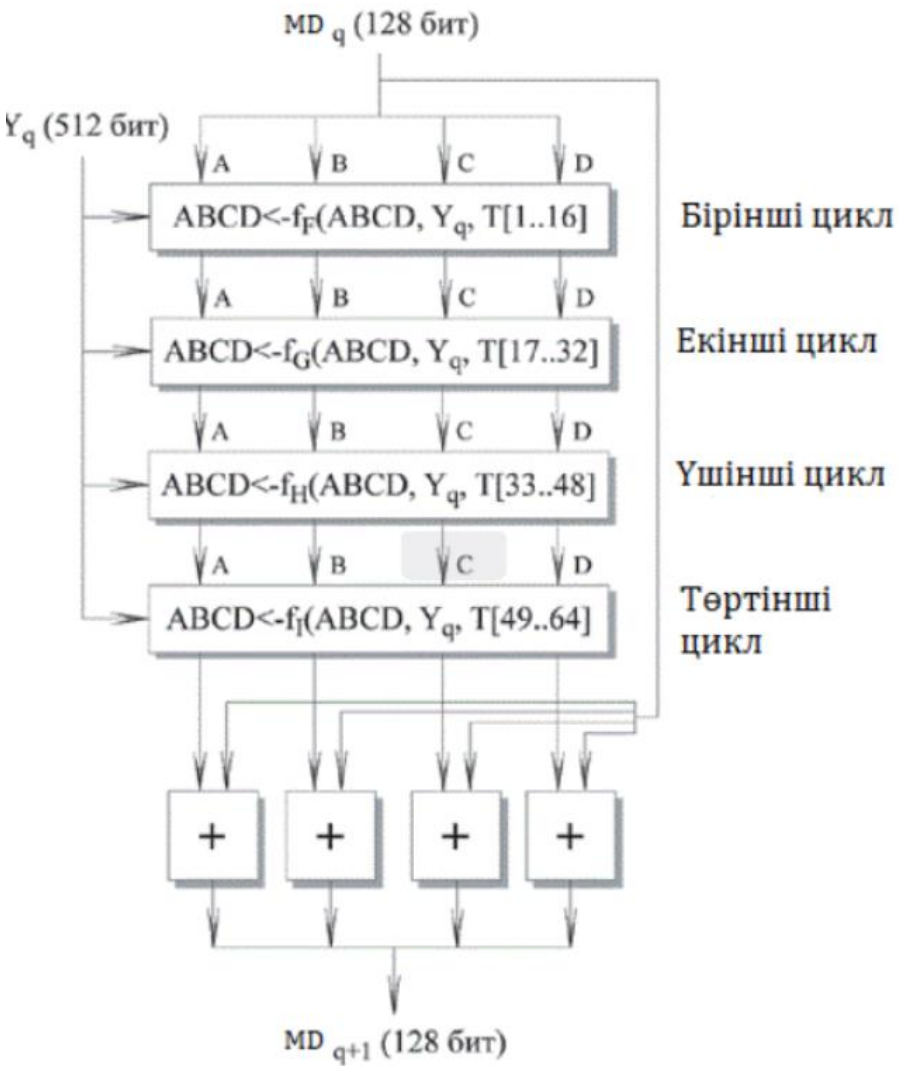
MD5 алгоритмі ақпаратты дайджест түрінде сақтаудан бөлек, соңғы мәнді тұздау және қайталану үдерісін қосуға мүмкіншілік береді. Бұл екі тәсілдердің көмегімен көптеген веб-қосымшалар қауіпсіздік деңгейін жаңа деңгейге жеткізуге жүзеге асырады. Қолданушы енгізген жеке ақпараттың соңғы мәндірінің алдына тұздау үдерісі орындалады, яғни дайджесттің алдынғы мәніне қосымша байттар енгізіледі. Нәтижесінде қолданушының жеке ақпараты жағымсыз шабуылдардан қорғай алатын қалыпқа ие болады.

MD5 дайджестерді тұздаудың негізгі екі түрлі тәсілмен жүзеге асыруға болады. Ол тәсілдерге бекітілген тұздау және айнымалы арқылы орындалатын тұздау тәсілдері жатқызылады.

Бекітілген тәсілдің орындалу үдерісінде дайджесті өңдеудегі байттардың тізбегі қосылады. Осы тәсілдің көмегімен құпия ақпаратты жасырын түрде сақтауға және оның бастапқы ұзындығын көбейту мүмкіндіктерге ие болады.

Айнымалы арқылы орындалатын тұздау әрбір өңделген соңғы мәндерге бөлек есептеуге және деректер қорында сақталған құпия сөздерді басқалардан құпия түрде сақтауға мүмкіндік беріп, ақпаратты қорғау үдерісін сенімді орындалуын қамтамасыздандырады.

Осы екі тәсілді аралас қолдану нәтижесінде тұздаудың соңғы мәні сегіз байттан көп көлемді алады. Дайджест құрамы құпия бөліктерден және кездейсоқ айнымалылардан тұратын тұздардан құралатының сипаттайды.

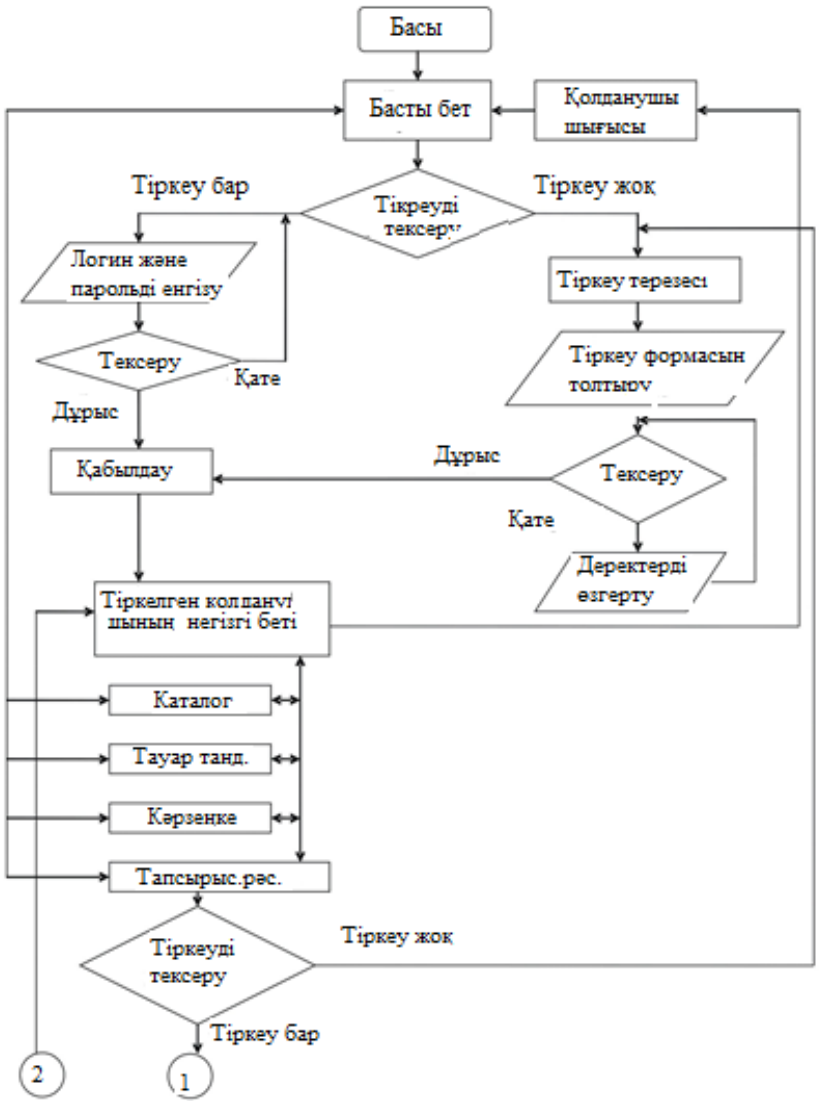


2.12 сурет – MD5 алгоритмінің блок сұлбасы

3 Веб-қосымшаның прототипі

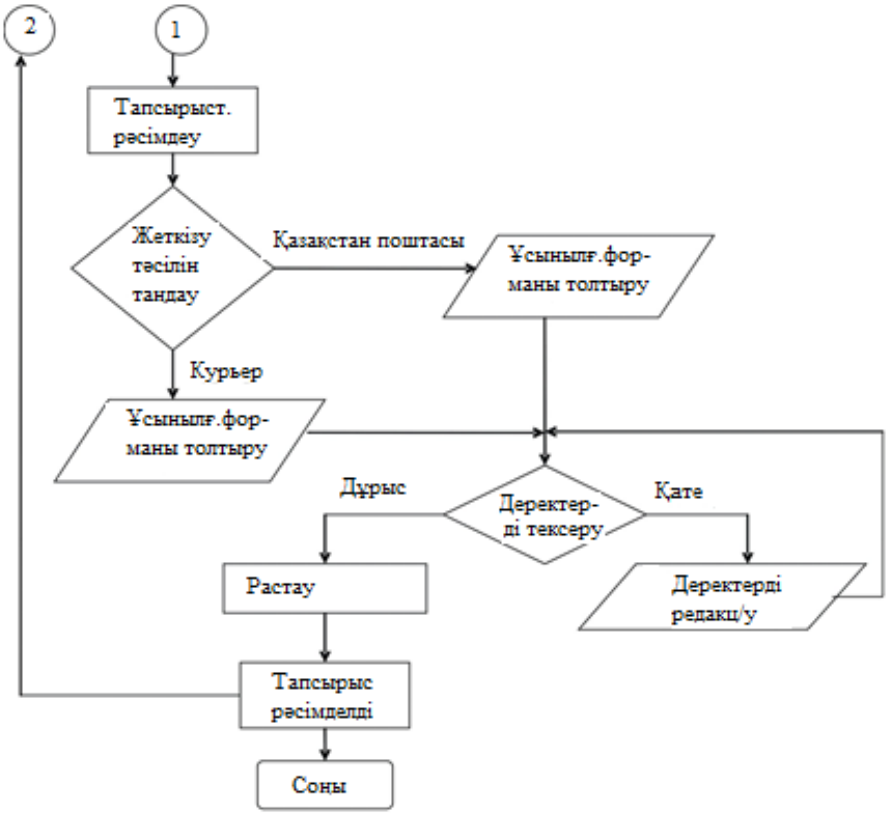
3.1 Веб-қосымшаның әзірлеу алгоритмі

Веб-қосымшаны құру алдында әзірлеуші веб-қосымша арқылы керекті қызметтермен әрекеттесуді қарастырады. Сондықтан әзірлеуші веб-қосымшаның жұмыс істеу алгоритміне негізделеді. Алгоритм бойынша қолданушыны веб-қосымша ішінде бастапқы қадамнан бастап жән соңғы қадамына дейін қолданушыны жүргізу керек. Алгоритмдегі қадамдар қолданушыны тексеру қадамдарынан өткізу арқылы қауіпсіздікті тексереді. Егер тексеру барысында қателіктер пайда болса, қолданушыны кері қадамға көшіреді. Толық қадамдардан өткен қолданушы керекті қызметке ие болады.



3.1 сурет – Әрекеттесу алгоритмінің бастапқы қадамы

Алгоритм қауіпсіздікті қамтамасыз ететін деңгейді қамтыған жағдайда, веб-қосымшаны прототипін құру үдерісі жүзеге асады. Прототип бойынша веб-қосымша бірнеше парақшалардан құрылады.



3.2 сурет – Әрекеттесу алгоритмінің аяқталуы

3.2 Веб-қосымша прототипінің элементтері

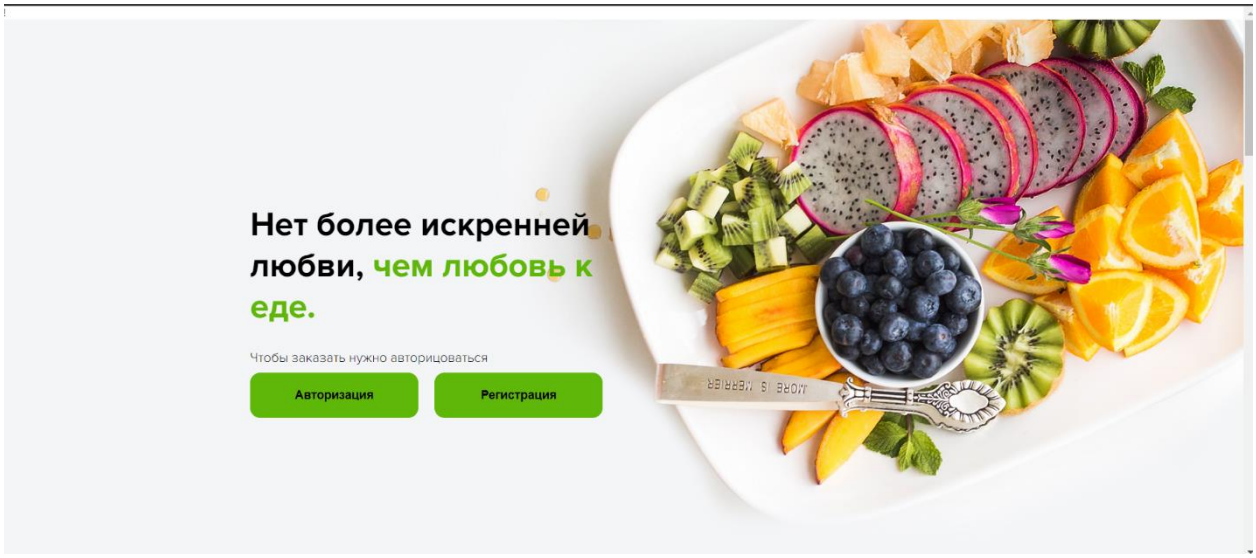
Басты парақшаларда қолданушы енгізген деректерді қауіпсіз түрде өңдеу үдерісін қамтиді.

Веб-қосымшаның прототипін құрайтын элементтер:

– басты парақша;
– авторизация парақшасы;
– тіркелу парақшасы;
– тауарлар бөлімі;
– таңдалған тауады төлем парақшасы.

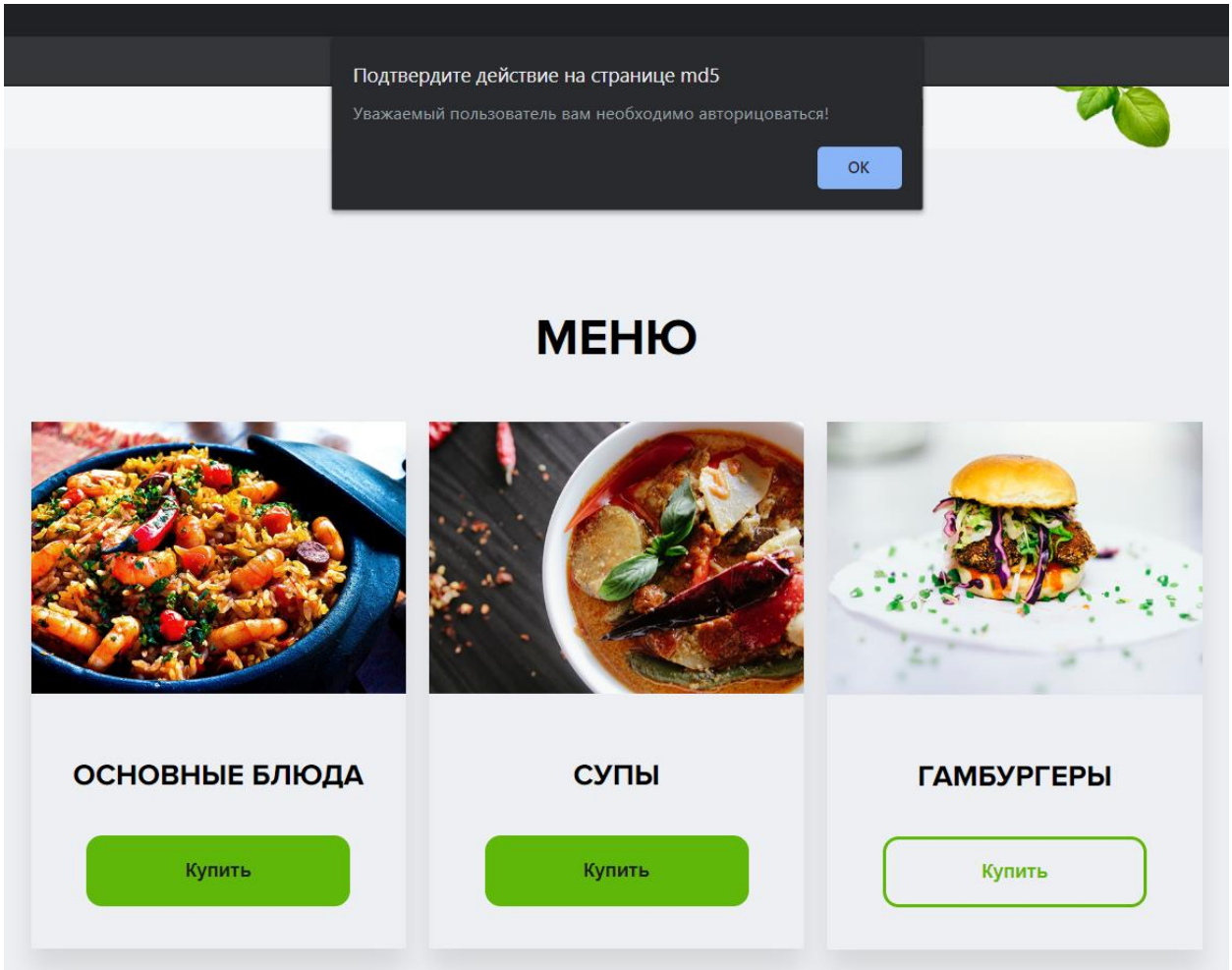
Басты парақшада веб-қосымша туралы негізгі мәлеметтер орналасады немесе қандай қызмет көрсету түрлерін бар екенің сипаттайды. Басты парақ 3 бөлімнен құралған.

Бірінші бөлімде қолданушы керекті қызмет түрін алу үшін тіркелу парақшасынан өту керек, егер тіркелмеген жағдайда веб-қосымша қолданушыға қызмет көрсетуден бас тартады.



3.3 сурет – Басты парақша көрінісі

Қолданушы тіркелмеген жағдайда ескерту хабарламасы жіберіледі. Хабарлама Javascript бағдарламалық тілінде қолданушыға көрсетіледі. Яғни javascript тіліндегі alert ішкі функциясы арқылы модальді терезе түрінде орындалады. Функция хабарлама ретіндегі сөздерді және тек қорытынды мәніне тек шындық параметрін қабылдайды.



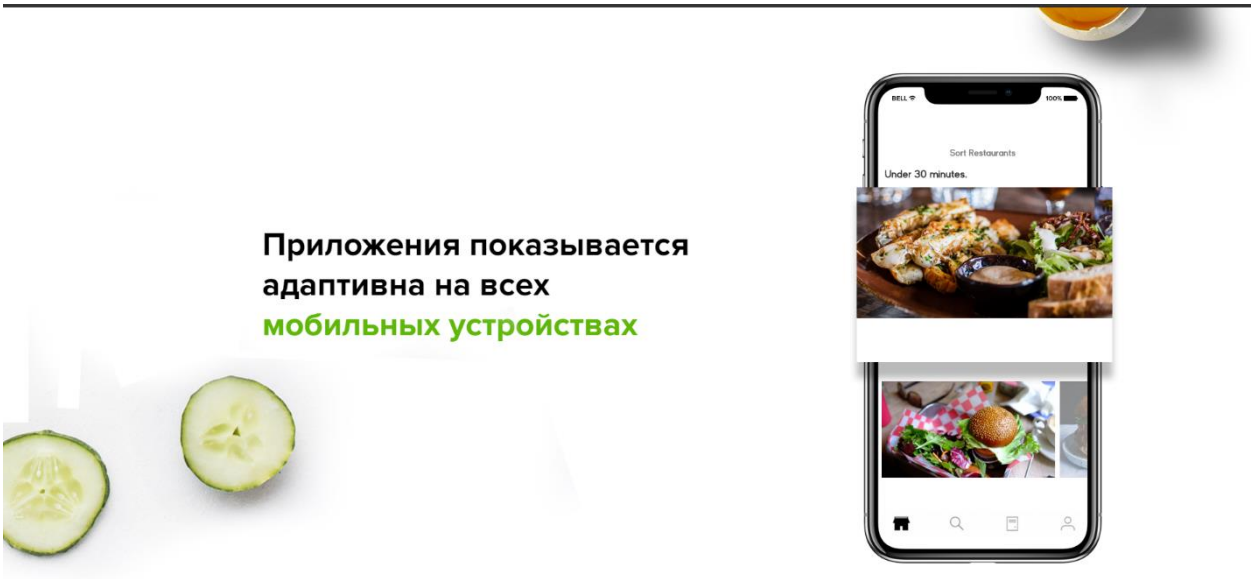
3.4 сурет – Тіркелмеген қолданушыларға хабарлама көрсету үдерісі

Екінші бөлімде тауарлардан құралған блоктар орналасқан. Алгоритм бойынша қолданушы тіркелген жағдайда тауарларды сатып алуға рұқсат беріледі. Веб-қосымша екі басты бөлімнен құрылған және үш кішкентай бөлімдерді құрайды.

```
<div class="menu-nur-block">
    <div class="menu-nur__item">
        <div class="menu-nur__img">
            <img src="img/menu-nur-img1.png" alt="menu-nur">
        </div>
        <div class="menu-nur__title">ОСНОВНЫЕ БЛЮДА</div>
        <button class="button menu-nur__get"><a href="oplata.php">Купить</a></button>
    </div>
    <div class="menu-nur__item">
        <div class="menu-nur__img">
            <img src="img/menu-nur-img2.png" alt="menu-nur">
        </div>
        <div class="menu-nur__title">СУПЫ</div>
        <button class="button menu-nur__get"><a href="oplata.php">Купить</a></button>
    </div>
    <div class="menu-nur__item">
        <div class="menu-nur__img">
            <img src="img/menu-nur-img3.png" alt="menu-nur">
        </div>
        <div class="menu-nur__title">ГАМБУРГЕРЫ</div>
        <button class="button menu-nur__get"><a href="oplata.php">Купить</a></button>
    </div>
</div>
```

3.5 сурет – Басты бөлімнің құрылымы

Үшінші бөлім мобильді құрылғылардың бейімделу туралы мәлімет көрсетіледі. Веб-қосымша қолданушылар санын арттыру үшін мобильді құрылғыларда бейімделу үдерісі ыңғайлы құрастырылады.



3.6 сурет – Мобильді құрылғылар бөлігі

Мобильді құрылғыларда биімделу арнайы @media сұраулар арқылы жүзеге асырылады. @media сұраулар CSS (Cascading Style Sheets – яғни каскадты кестені стильдеу) тілімен тікелей жұмыс істейді. Сұрау жіберу

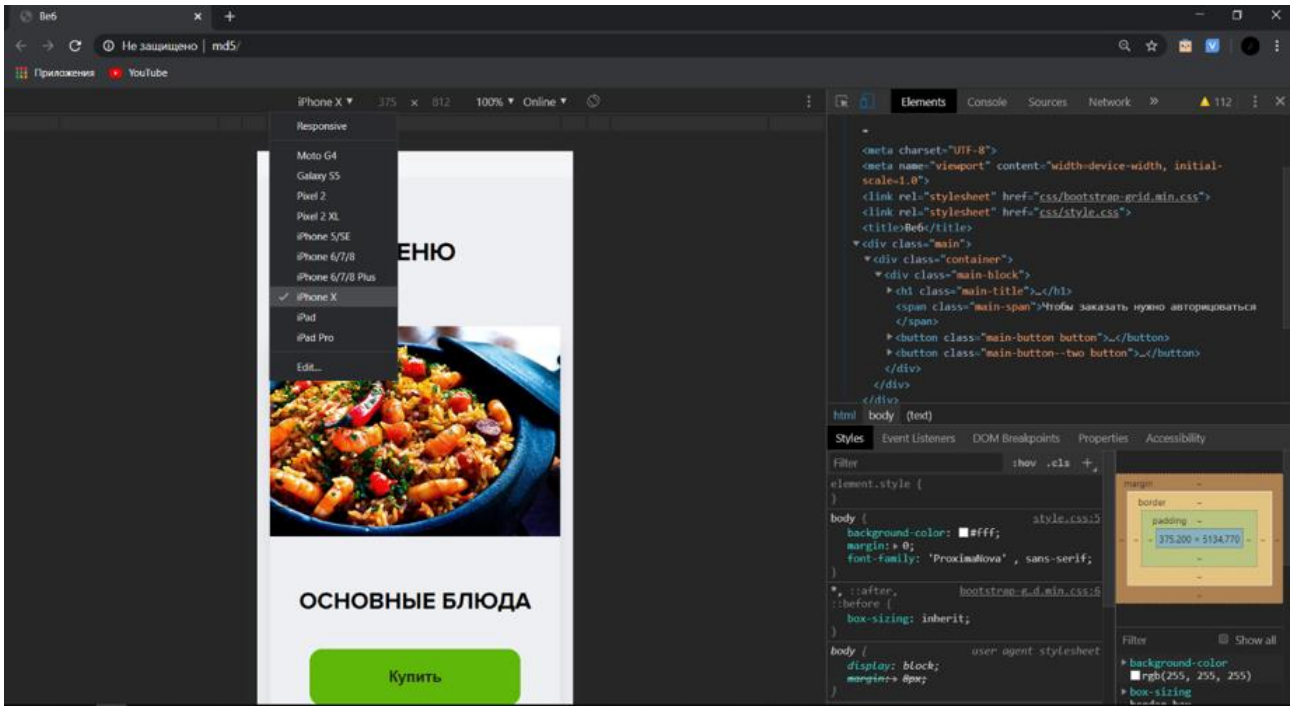
барысында қолданушының веб-шолғышындағы қабылдайтын форматтарын өндейді. Егер жіберілген сұраулар және қабылдайтын форматтар арасында қателіктер болмаса @media қосылады.

```
@media(min-width: 320px) and (max-width: 768px){
.main-content{

  padding: 70px 0;
}
.global-form{
   width: 300px;
   background-color: rgba(255, 255, 255, 0.7);
}
.global-input{
  padding: 20px 10px;
  width: 250px;
  margin-bottom: 20px;
  font-size: 15px;
}
}
```

3.7 сурет – @media сұрауларды қосу

Мобильді құрылғылардың көрсету ені әр түрлі болғандықтан бейімделу үдерісі ұзақ орындалады. Әрбір құрылғылардың көрсетуіне байланысты веб-шолғыш пиксель түрінде енің санайды. Егер берілген форматты қабылданған жағдайда қолданушы құрылғысында элементтер өзгереді.



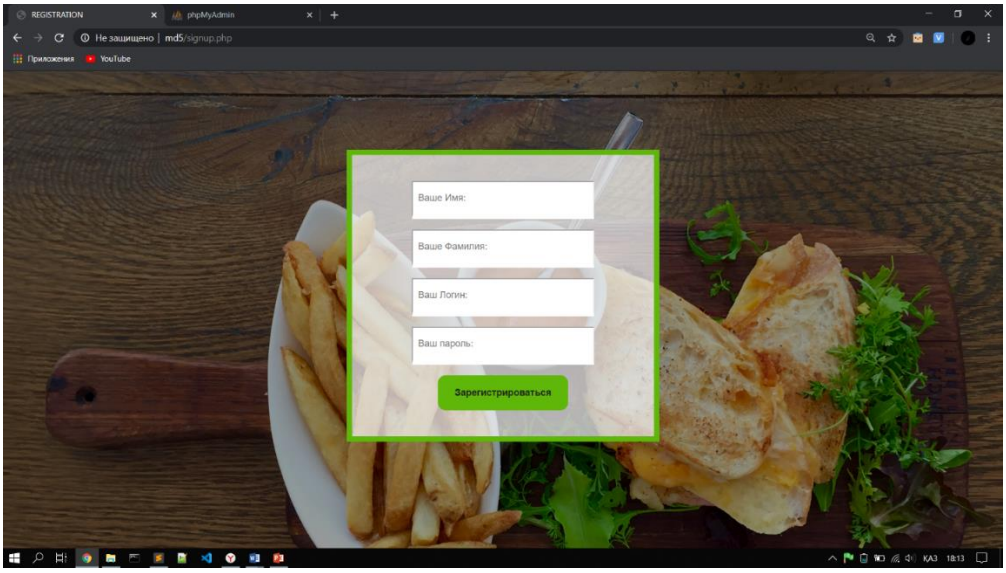
3.8 сурет – Веб-қосымшаның Iphone X құрылғысында бейімделу көрінісі

Көптеген веб-шолғыштар Moto G4 – ден бастап Iphone X – ке дейінгі мобильді құрылғыларды қабылдайды.

3.3 Веб-қосымшаның жұмыс істеу интерфейсі

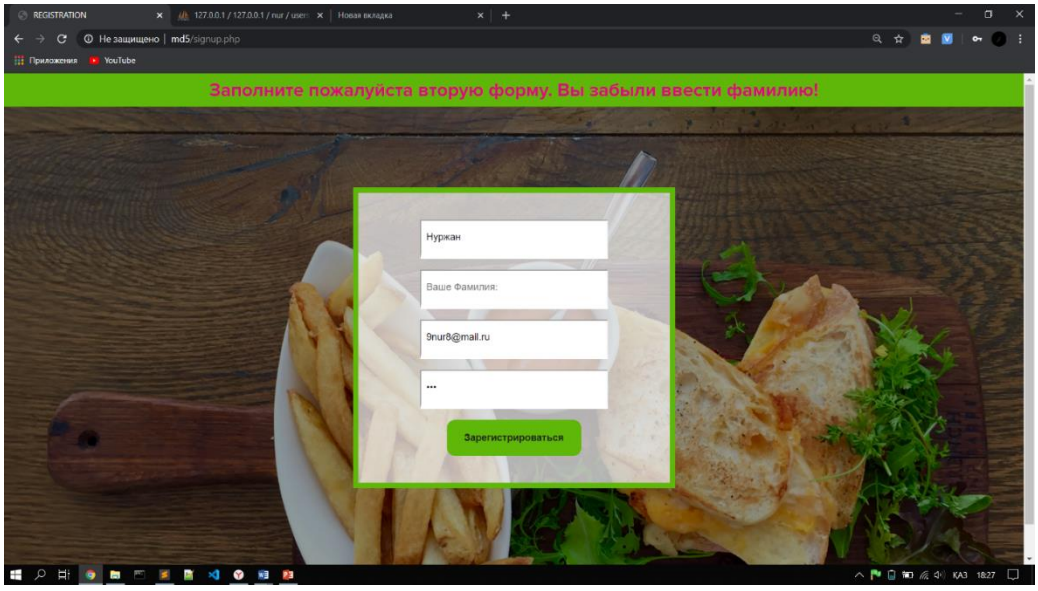
Қолданушы ең алдымен веб-қосымша интерфейсін қолдану негізде өзіне қажет қызмет түрлерін алады. Веб-қосымшаның интерфейсінің ыңғайлығының нәтижесі ретінде қолданушының веб-қосымшаны жиі қолдануды жатқызуға болады.

Бұл қосымшада қолданушы веб-интерфейстін жұмысымен әрекеттесу үдерісін бақылауға болады. Қолданушы веб-қосымшаға кірген сәтте бірінші басты парақшаны көреді, егер қолданушы тіркелмеген жағдайда басты парақша сілтемеге нұсқайды. Қауіпсіздікті қамтамасыздандыру кезеңдерінің бірі тіркелу парақшасында жүзеге асырылады. Тіркелу барысында қолданушы толтырған ақпараттар тексеру үдерісінен өткізіледі.



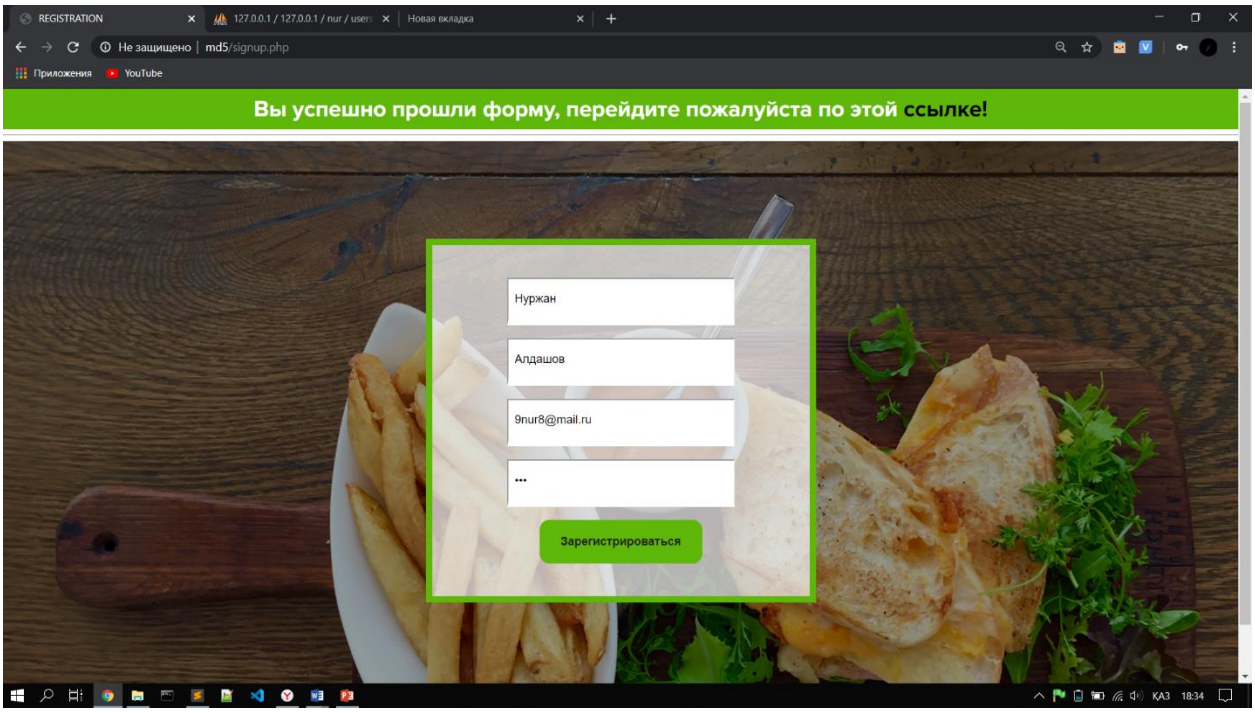
3.9 сурет – Тіркелу парақшасы

Қолданушы енгізген ақпараттар әрбір input-қа толтырған немесе толтырмай енгізілген деректерді тексереді.



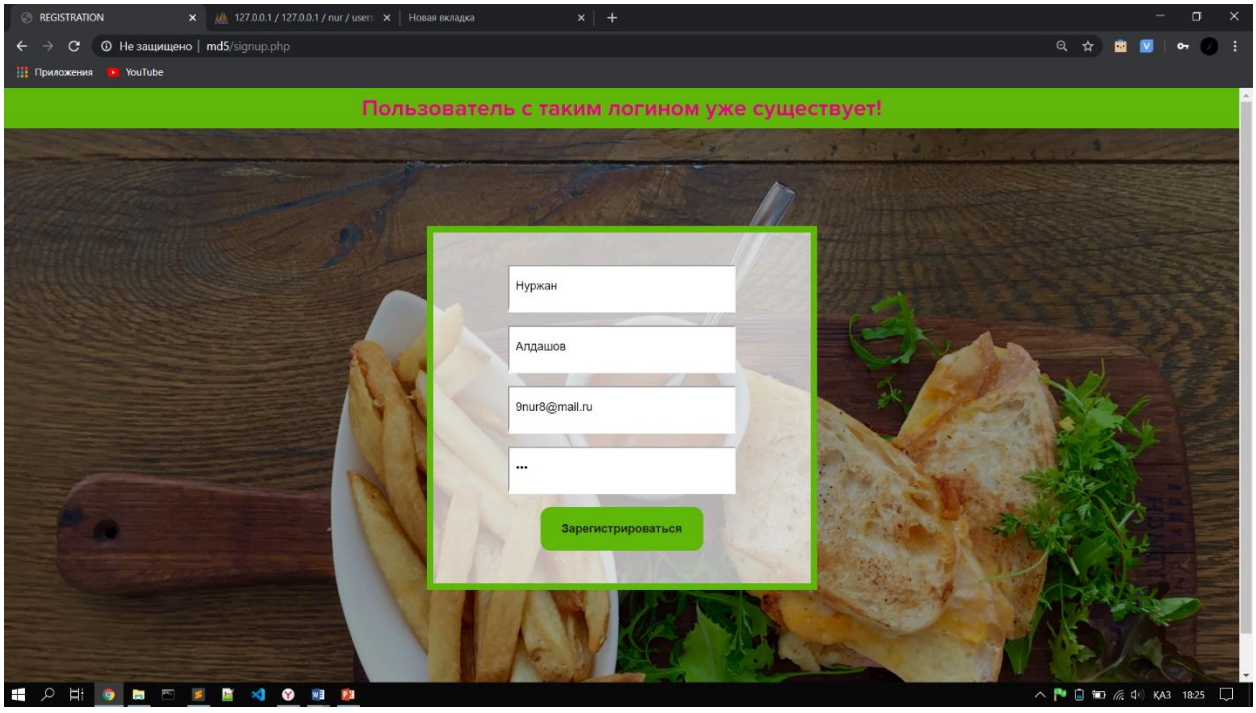
3.10 сурет – Енгізілген ақпаратты тексеру

Тіркелу формасын қолданушы толыйтай толтырмағанынша келесі қадамға көшірілмейді. Сондықтан тіркелу формасын берілген стандартқа сай толтыру қажет. Мысалы:



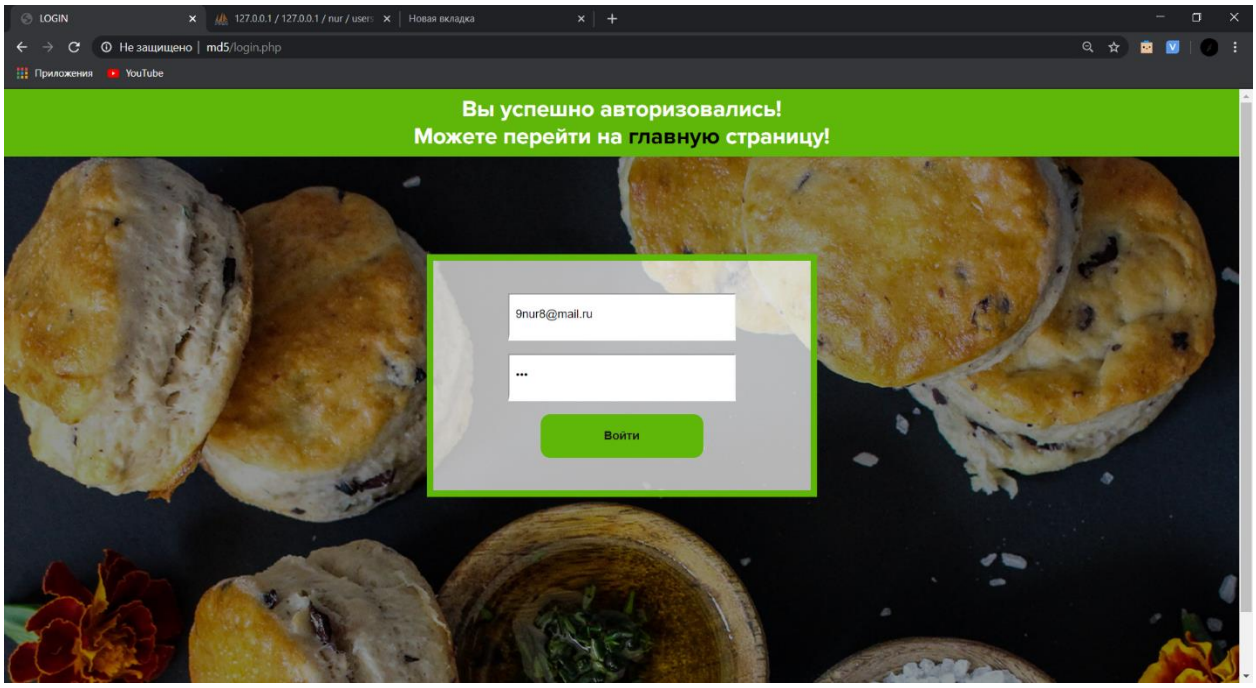
3.11 сурет – Толық толтырылған тіркелу форма

Егер шабуылдаушы қолданушының атынан кіретін жағдайды қарастырса, веб-қосымша тексерулердің арқасында шабуылдаушыны жүйеге кіргізбейді. Өйткені жүйе логинді салыстыру нәтижесінде жүйеге кіргізбеу туралы шешім қабылданады.



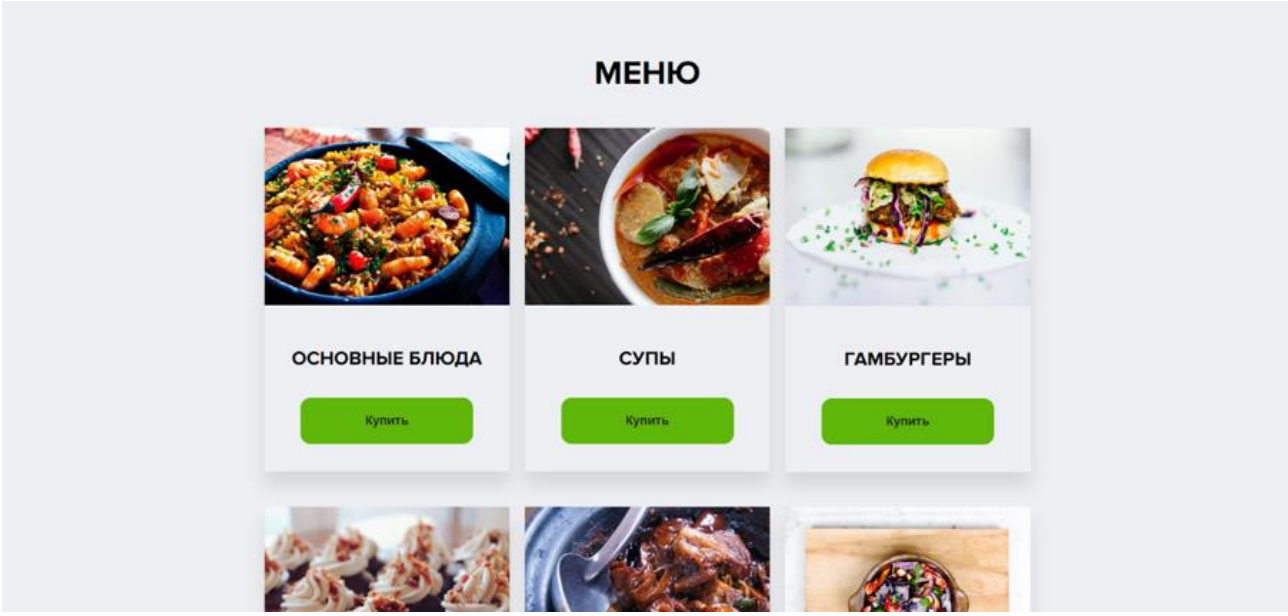
3.12 сурет –Логиндердің сәйкестік тексеру үдерісі

Егер тіркелу үдерісін сәтті өткен жағдайда веб-қосымша авторизация парақшасынан тексеру қадамы іске қосылады. Қадамнан өткен қолданушы веб-қосымшада тауарды сатып алуға рұқсат беріледі. Авторизация парақшасын толтыру:

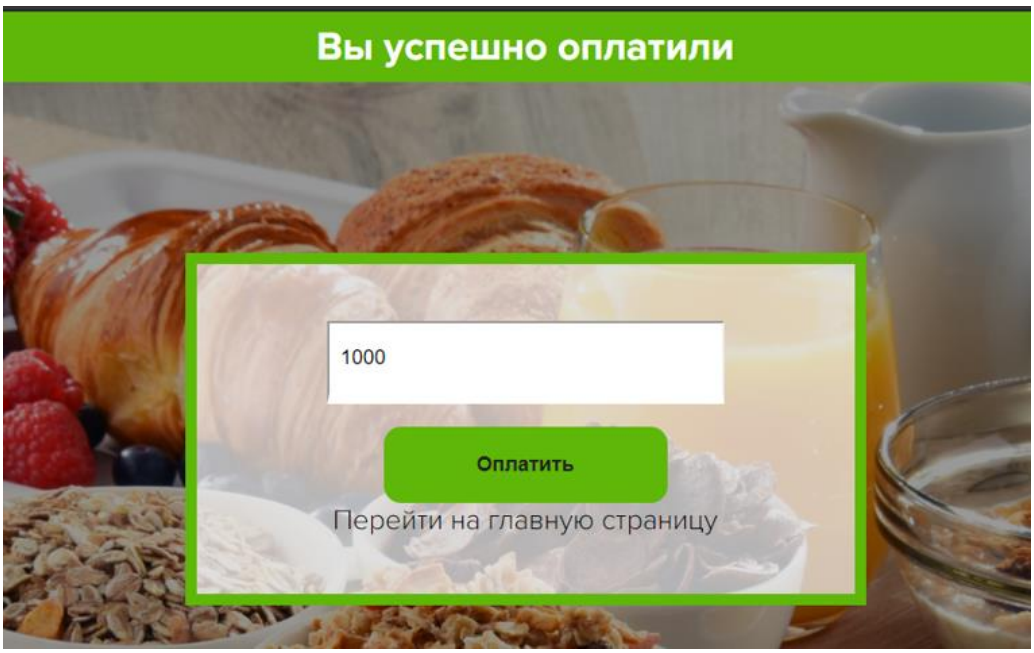


3.13 сурет – Авторизациядан сәтті өту үдерісі

Сонғы қадам орындалу барысында қолданушыны басты парақшаға сілтейді. Веб-қосымшаның басты парақшасында қолданушы тауар сатып алуға жүйе толық рұқсатын береді. Қолданушы тандаған тауардың «Сатып алу» батырмасын басқаннан кейін, қолданушыны төлеу парақшасына көшіреді.



3.14 сурет – Тауарлар парақшасы



3.15 сурет – Төлем сәтті орындалды

Егер төлем сәтті орындалған жағдайда деректер қорына түседі. Қолданушының сұраулары ең алдымен серверден өткізіліп, деректер қорына әкімші атындағы жүйе арқылы жүзеге асырылады. Сол-себебті деректер қорына қосылу үдерісі бірінші орындалуы қажет. Ол үшін db.php файлында деректер қорына дейінгі толық орналасқан жері жазылады. Құжатта міндетті түрде деректер қорының порт нөмірі, деректер қорының аты және әкімшінің атын көрсетіледі.

```
signup.php | db.php | logsuc.php | login.php | index.php

1  <?php
2  require "libs/rb.php";
3  R::setup( 'mysql:host=127.0.0.1;port=3306;dbname=nur','root');
4  session_start();
5  ?>
6
```

3.16 сурет – MySQL деректер қорына қосылу

Жергілікті сервермен және деректер қоры арасында байланыс ақаулар туындамаған жағдайда, қолданушының сұраулары деректер қорында ақы төлеу кестесіне енгізілуі тиіс.

3.17 сурет – Төлемнің деректер қорына түсуі

3.4 Веб-қосымшаның қорғану алгоритмі

Веб-қосымшаны қорғау алгоритмі қолданушы енгізген ақпаратты шифрлау арқыла жүзеге асады. Шифрлау үшін әртүрлі алгоритмдер колданылады, алайда қауіпсіздікті қамтамасыздандыру үдерісінде қолданылатын алгоритмдердің бірі Message Digest 5 немесе MD5 алгоритмі.

Алгоритм бойынша қолданушы енгізген құпия ақпарат тексеру циклдарына өткізіліп, деректер қорына берік қорғалған деректер жіберіледі. Циклдің жұрыс үдерісі 4 шартты қабылдауды мәжбүр етеді. Егер циклдағы шарттын дұрыс мәнді қабылдамаған жағдайда, қайта тексеріс кезеңдеріне көшіріледі. Осындай шарттарды толық қамтыған қолданушының ақпараты серверден рұқсат алып деректер қорына енгізіледі.

signup.php

```php
<?php
 require "db.php";

 $data = $_POST;
 if(isset($data['do_signup']))
 {
   //здесь регистрируем

   $errors = array();
   if(trim($data['name']) == '')
   {
      $errors[] = 'Введите Имя!';
   }

   if(trim($data['fam']) == '')
   {
      $errors[] = 'Заполните пожалуйста вторую форму. Вы забыли ввести фамилию!';
   }

   if(trim($data['login']) == '')
   {
      $errors[] = 'Введите ваш логин!';
   }
   if($data['password'] == '')
   {
      $errors[] = 'Введите пароль:!';
   }

   if( R::count('users', "login = ?", array($data['login'])) > 0 )
   {
      $errors[] = 'Пользователь с таким логином уже существует!';
   }
```

3.18 сурет – Қолданушы енгізген ақпараттарды тексеру

3.5 MD5 шифрмен парольдік қорғау

Қорғалған веб-қосымшаны әзірлеу үшін құпия ақпаратты сенімді сақтау мәселесі бірінші кезекте қаралу қажет. Сенімді қосымша сервердегі деректер қорында сақталатын жеке ақпараттың қауіпсіздігін қамтамасыздандырады. Сол-себебті MD5 шифрлау алгоритмі кең қолданылады. Бұл шифрлау алгоритмі деректер қорында сақталған құпия ақпаратты бір жақты хэштеу арқылы қауіпсіздік деңгейін жоғары дәрежеде сақтауға негіз болады. Веб-қосымшаның деректер қорына әкімші рұқсаты бар қолданушы кірген жағдайда,

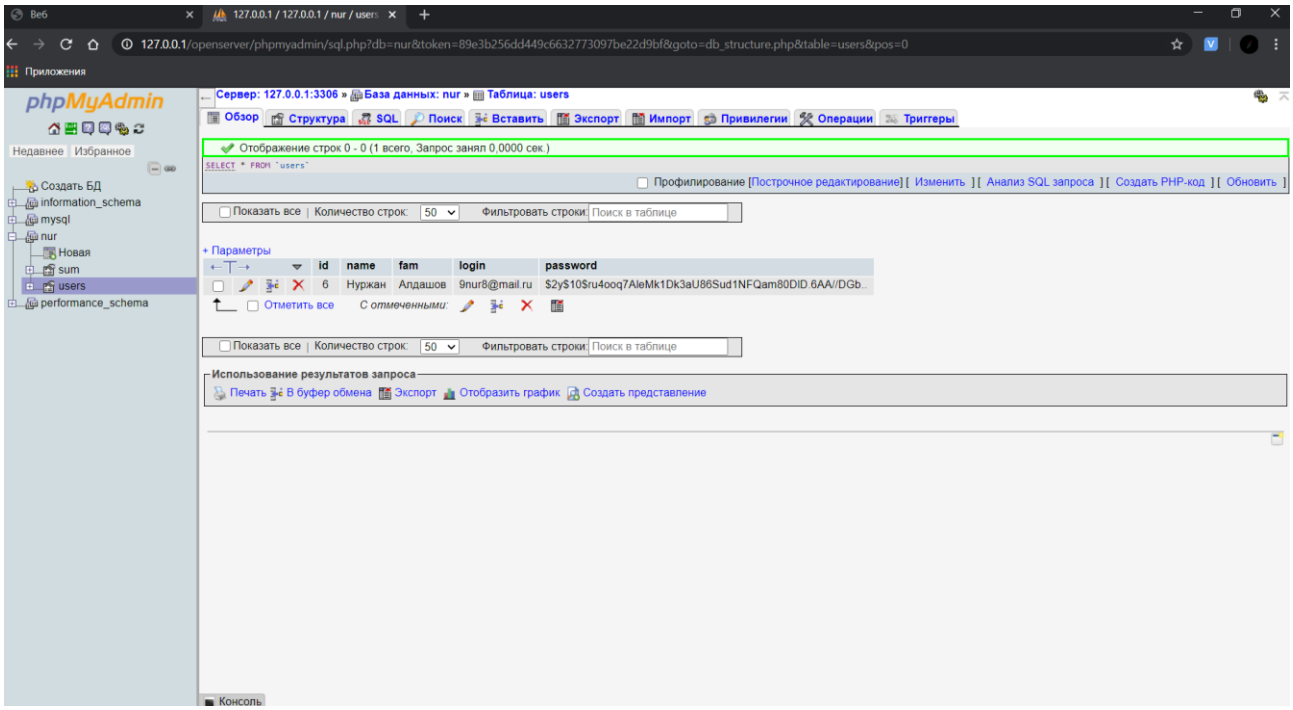
ол құпия ақпаратты кері шифрлай алмайды, яғни бұл үдеріс бастапқы енгізілген ақпарат әрқашан бір шифрланған мән ретінде сипатталады. MD5 жұмыс істеу үдерісі барысында дайджесттерді бір-бірімен салыстыру тәсілі арқылы қолданушы енгізген және деректер қорында сақталған ақпартты талдайды. Осындай салыстырулардың нәтижесінде жеке ақпаратты тек қана қолданушы білуін қамтамасыздандырады. MD5 шифрлау алгоритмімен парольдік қорғауды жүзеге асыру үдерісі келесі фрагментте көрсетіледі:

```
if( empty($errors) )
        {
                $user = R::dispense('users');
                $user->name = $data['name'];
                $user->fam = $data['fam'];
                $user->login = $data['login'];
                $user->password = md5($data['password']);
                R::store($user);
                echo '<div class="message">Сіз сәтті формадан өттіңіз, осы сілтеме бойынша <a href="/login.php">өтіңіз!  </a></div><hr>';
        } else
        {
                echo '<div class="message error">'.array_shift($errors).'</div>';
        }
```

Жоғарыда көрсетілгендей, ең алдымен қолданушы енгізген ақпаратты шарттардан өткізу арқылы тексеріс үдерісі жүзеге асырады. Әр айнымалы өзіндік массивтерді түрінде реттеріліп, бір жақты хештеу кезеңіне көшіріледі. Егер осы шарттардың бірінде қателіктер орындалса, цикл бастапқы мәндегі бірінші раундқа сілтемелейді. Шарт бойынша қолданушы енгізген ақпарат айнымалыларда массив түрінде деректер қорында жіберіледі. Келесі фрагметте қолданушы терген ақпараттың серверда сақтау механизмін жүзеге асырылады:

```
<?php
 require "db.php";
 $data = $_POST;
 if( isset($data['do_login']) )
 {
        $errors = array();
        $user = R::findOne('users', 'login = ?', array($data['login']));
        if( $user)
        {
                if(password_verify($data['password'], $user->password)){
                        $_SESSION['logged_user'] = $user;
                        echo '<div class="message">Сәтті өттіңіз!<br/>
                        Келесі <a href="logsuc.php">басты </a> парақ</div>';
                }else{
                        $errors[] = Құпия сөз қате!';
                }}else
```

```
		{
			$errors[] = 'Қолданушы осындай логинмен табылған жоқ';
		}
	if(! empty($errors) )
	{
		echo '<div class="message error">'.array_shift($errors).'</div>';
	}
}
?>
```



3.19 сурет – MD5 шифрмен парольдік қорғау

Деректер қорындағы кестеге сілтеме жасау әсерінен серверге жүктеме аз көлемеде сезіледі. Нәтижесінде веб-қосымшаның қолданушы жіберген сұрауларды өңдеу жылдамдығы артып, қолданушының құпия ақпараты шифрланған түрде түседі.

4 Ақпараттық қауіпсіздік тәуекелдері

4.1 Тәуекелдерді бағалау және анализ

Дипломдық жұмыстың осы бөлігінде веб-қосымшаның ақпараттық қауіпсіздікті қамтамасыз ету мақсатында тәуекелдерін бағалаймыз. Жоба тәуекелі – бұл пайда болған жағдайда жобаның мақсаттарының біріне оң немесе теріс әсері бар белгісіз оқиға[12].

Кез келген тәуекелдің екі параметрі болады: әсер ету және туындау ықтималдығы.

Әсер ету мәндерін және тәуекелдің туындау ықтималдығын анықтау үшін 0-ден 1-ге дейінгі шкала пайдаланылады:

0 - оқиға нақты болмайды;

1 - оқиға дәл болатынын біледі.

Дипломдық жұмысты әзірлеу барысында қорғауды талап ететін активтер тізімі жасалды.

4.1 кесте – Активтер тізбесі

№	Активтер	Сипаттама	Саны
1	Web-сервер	Қолданушының сұрауларын өңдеу үшін арналады	1
2	CMS WordPress	Веб-қосымшаны басқаруға арналған платформа	1
3	Бастапқы код	Веб-қосымшасының логикалық жұмысын жүзеге асыруға арналған	-
4	Деректер қоры	Қолданушылардың жеке ақпараттарын сақтауға арналған БҚ	1
5	Веб-әзірлеуші	Веб-қосымшаны әзірлейтін маман	1

4.2 Есептік бөлім

4.2.1 Екі параметр бойынша бағалау әдісі

Екі параметр бойынша бағалау әдісі қауіптің туындау ықтималдығын бағалауды және залалды немесе әсерді бағалауды қамтиды. Осы әдістеме бойынша тәуекел 4.1 формуламен анықталады:

$$\text{Тәуекел} = \text{Пайда болу ықтималдығы} \times \text{Әсері} \qquad (4.1)$$

Бұл әдіс үш кезеңді қамтиды:

1) тәуекелдердің бастапқы есебі;

2) қолайсыз тәуекелдер үшін шараларды айқындау;

3) қайта есептеу.

4.2 - кесте – Тәуекелдерді бағалаудың қорытынды кестесі

Актив	Қауіптер	Осалдық	Әсері	Ықтималдығы	Тәуекелдің ең жоғары деңгейі	Тәуекелді өңдеу шаралары	Тәуекелдің қалдық деңгейі	Күні	Ескертпе
Web-сервер	Ақпаратқа рұқсатсыз қол жеткізу	Сессия таймаутының болмауы	2	4	8	Трафикті сүзу функциясын модульдерге орнату	4	4.06.2020	Сервермен үйлесімді нұсқаны орнату керек
CMS Word Press	Құпия сөзді автоматты таңдау үдерісі.	Сұраныстарды өңдеу үдерісінде уақыт бойынша шектеудің болмауы	3	3	9	Сұрауларды өңдеу кезінде құпия сөзді шифрлау және уақытқа шектеу қою	3	6.06.2020	Шифрлау символдарды қабылдап және уақытқа шектеуді қолмен баптау қажет
Бастапқы код	Мазмұнды өзгерту	Мазмұнды өзгертуге тыйым салатын	2	2	4	Қолжетімділікті шектеу, парольдік қорғау	0	8.06.2020	Сәйкестендіру кодын дұрыс

		индекстеу плагиндердің болмауы							таңдау керек
Дерек тер қоры	Жабдықтың істен шығуы	Ресурстарды дұрыс бөлмеу	2	3	6	Резервтік қалпына келтіру жүйесін баптау	3	4.06.2020	Сақтық көшірме үшін күн ауқымын таңдау керек
Веб-әзірле уші	Бағдарламал ық қате	Қызметкердің біліксіздігі, құралдардың дұрыс жұмыс істемеуі	2	2	4	Қызметкерлерді мерзімді оқыту	2	1.06.2020	Жиі тексерісте рді қамтамасы здандыру

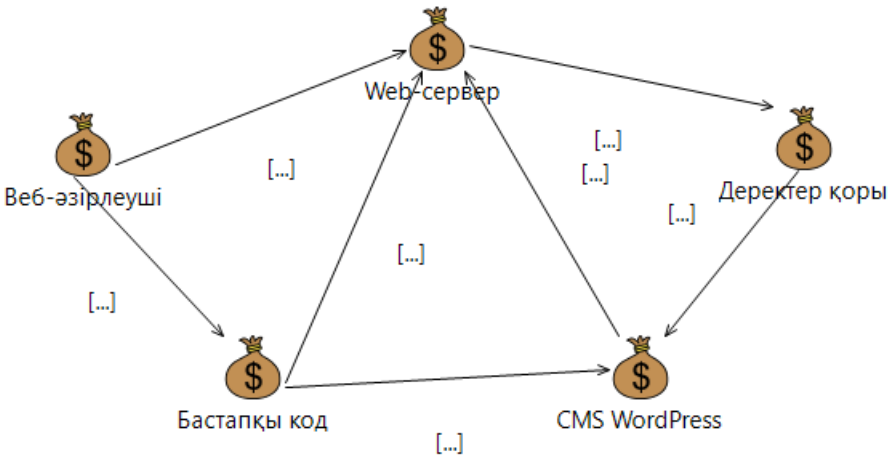
4.2.2 CORAS құралымен тәуекелдерді талдау

Coras әдіснамасы қауіпсіздік тәуекелдерін талдауға арналады. Жұмыс барысында тәуекелдер мен қатерлерді модельдеу үшін қолданылады.

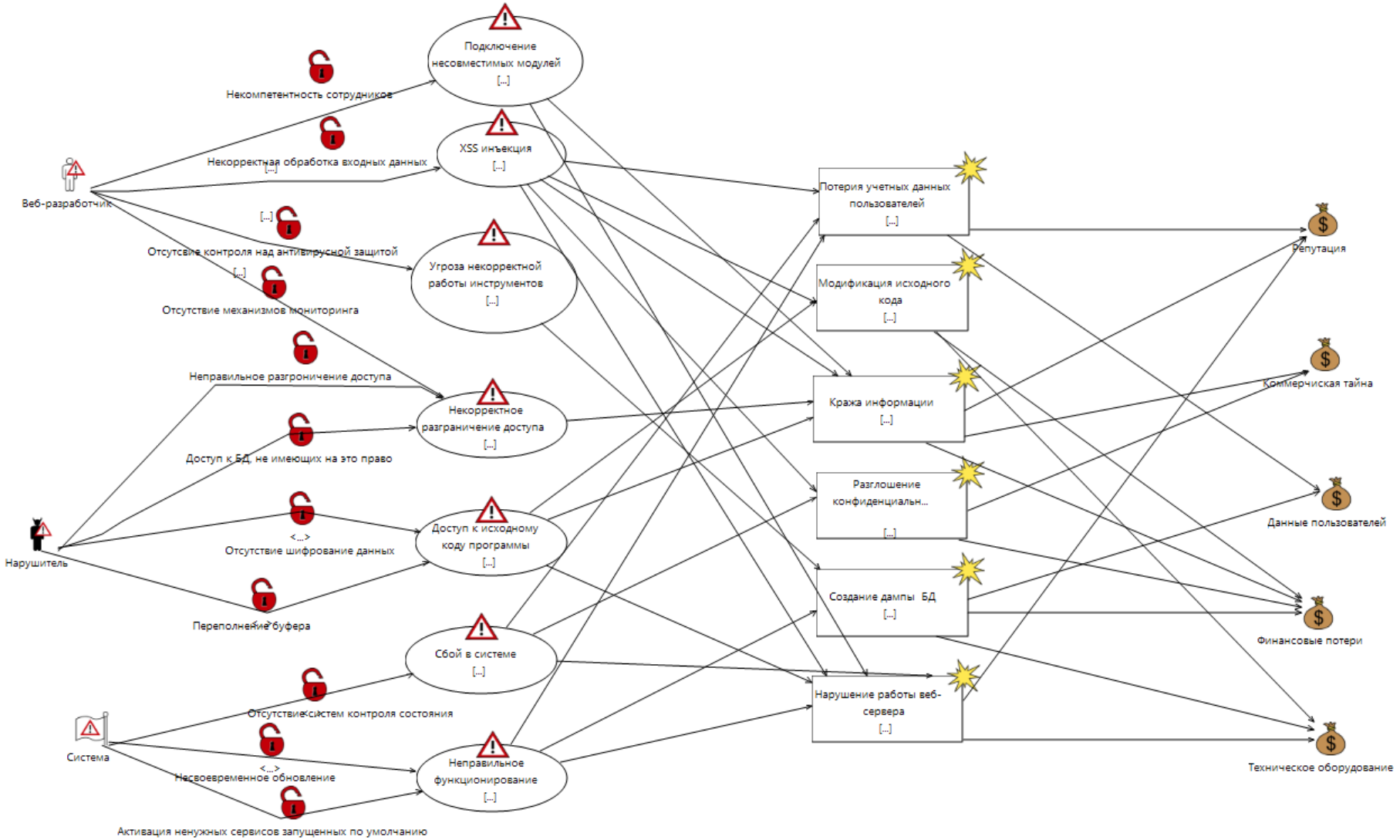
Coras UML-графикалық сипаттау тілін қолдану арқасында объектілі модельдеуді бағдарламалық қамтамасыз ету саласында қолданады. UML-графикалық сипаттау тілі бағдарламалық қамтамасыз етудегі анықтау, визуализация, жобалау және құжаттау үшін жасалды.

CORAS методологиясы Information Society Technologies бағдарламасы аясында әзірленген. Оның мәні Event-Tree-Analysis, Марков тізбегі, HazOp және FMECA сияқты тәуекелдерге талдау жүргізу әдістерін бейімдеу, нақтылау және біріктіруден тұрады.[13]

Дипломдық жобада ақпараттық қауіпсіздік тәуекелдерін талдау үдерісінде қолданылған активтерді пайдалану нәтижесінде активтер диаграммасын салдық. Бұл активтер 4.1 суретте көрсетіледі.

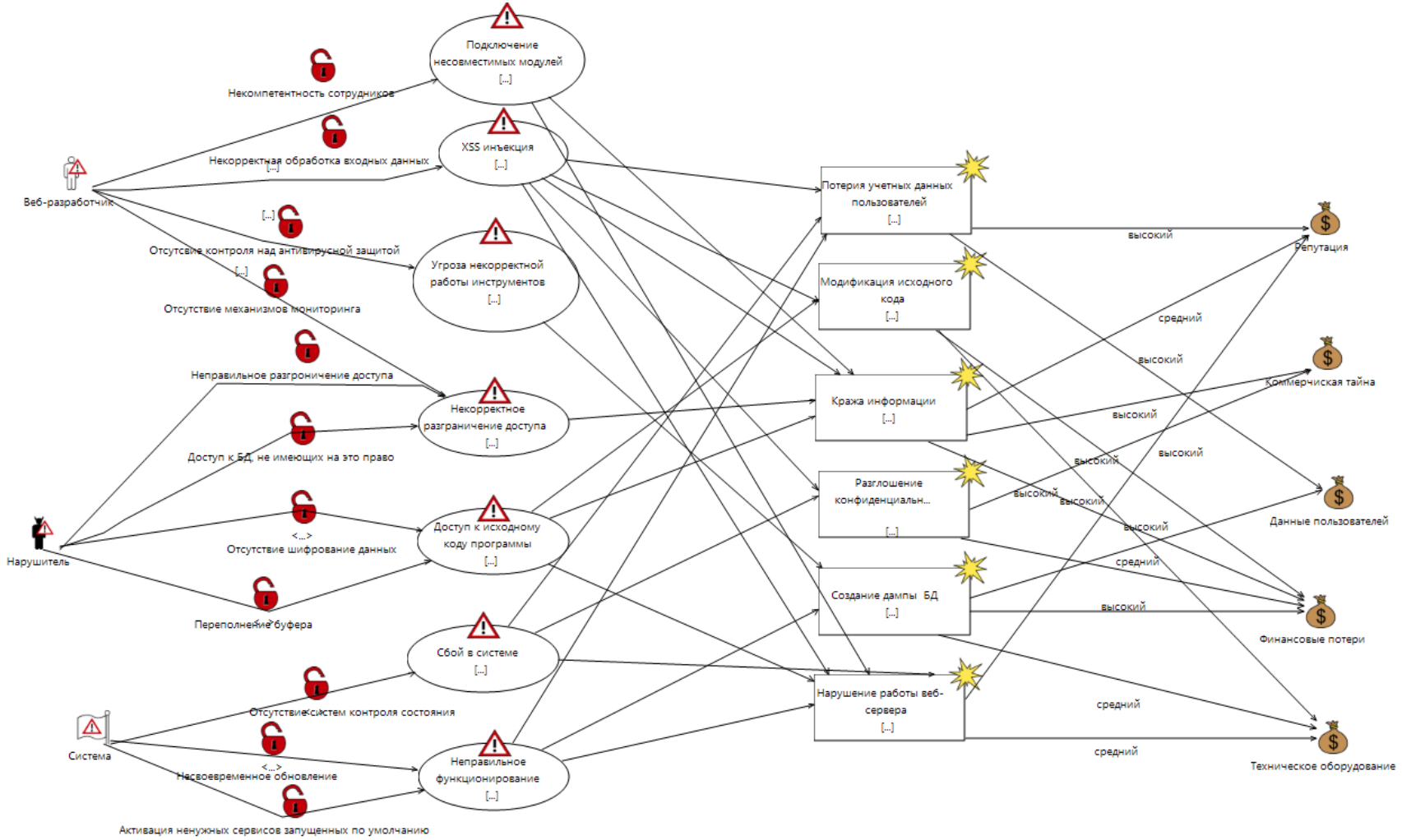


4.1 сурет – Активтер диаграммасы



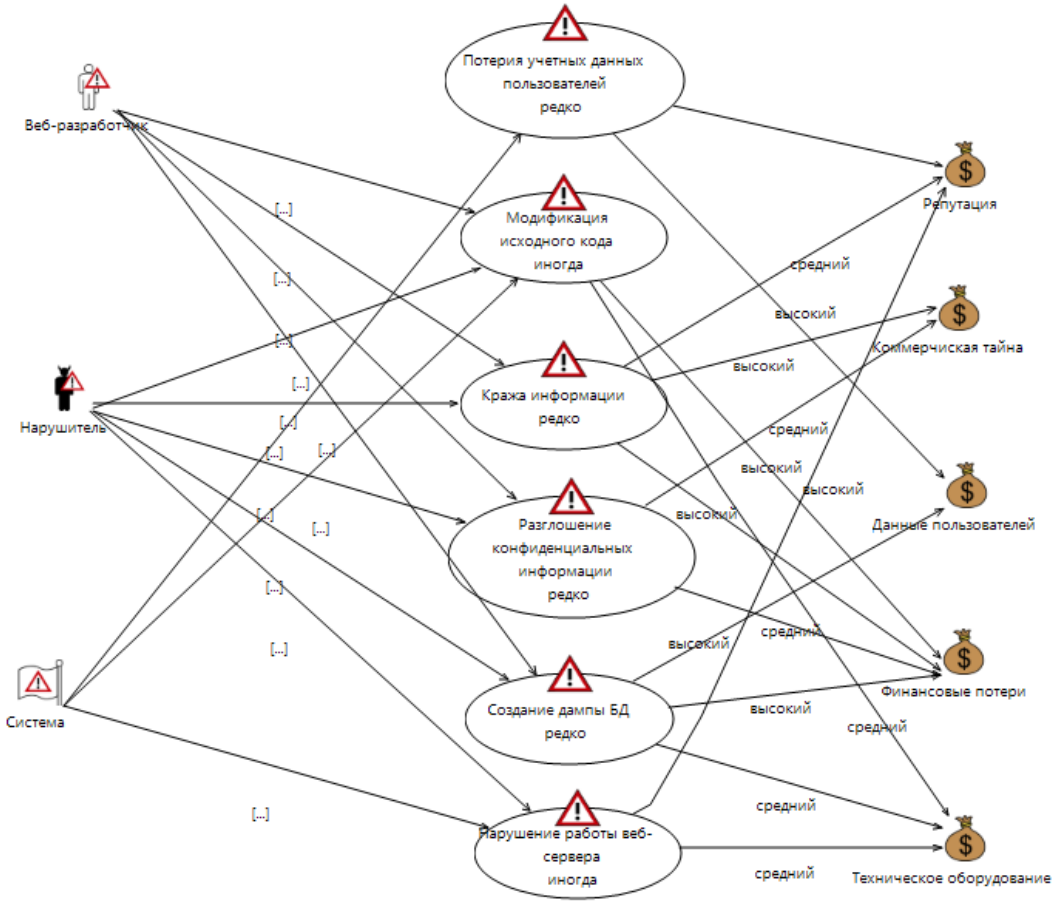
4.2 сурет – Қауіптер диаграммасы

Келесі суретте жүзеге асыру ықтималдығымен қауіптер моделі салынады.



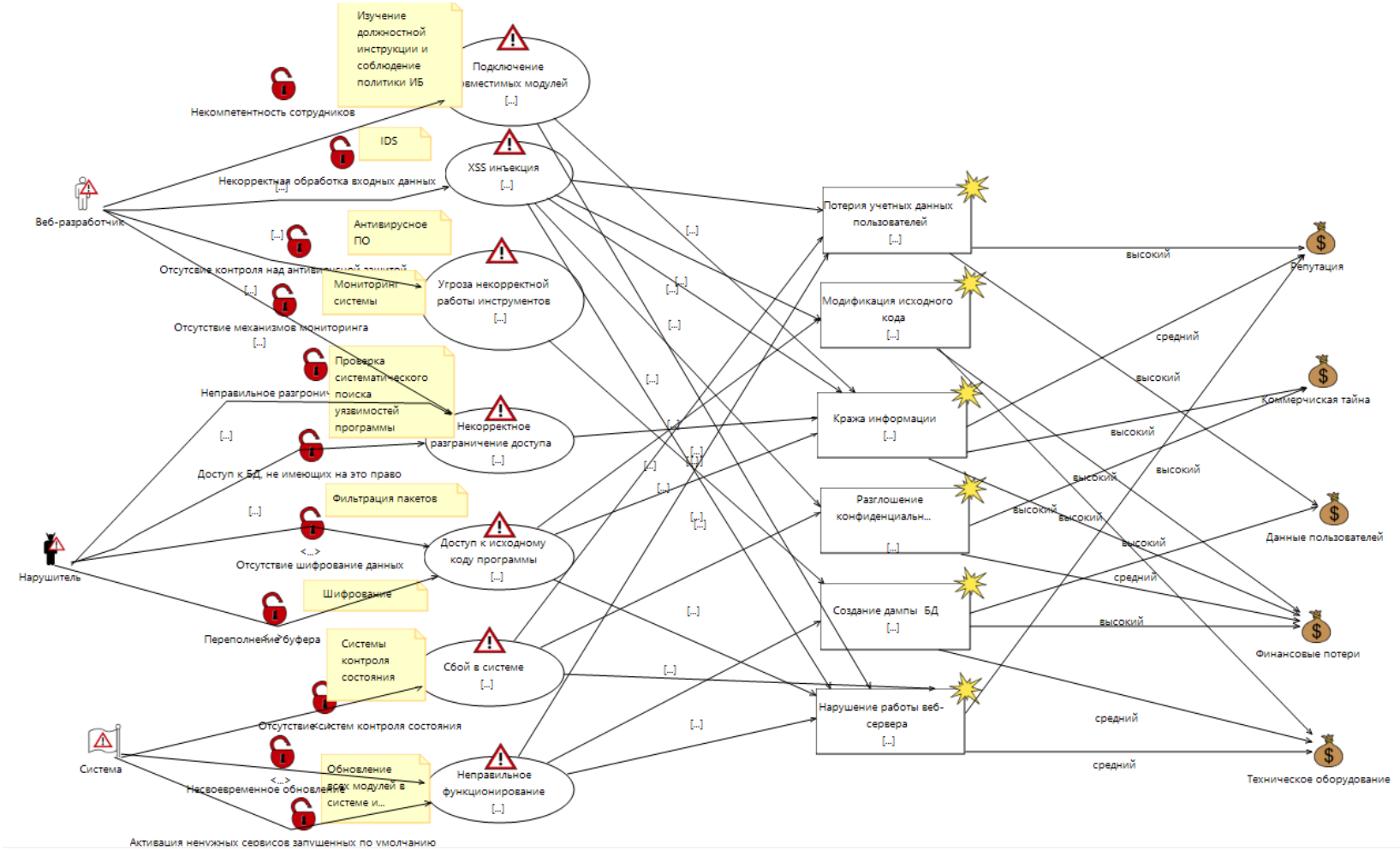
4.3 сурет – Жүзеге асыру ықтималдығымен қауіптер моделі

Қандай тәуекелдер қолайсыз екенін анықтау үшін тәуекелдер диаграммасын құру қажет.



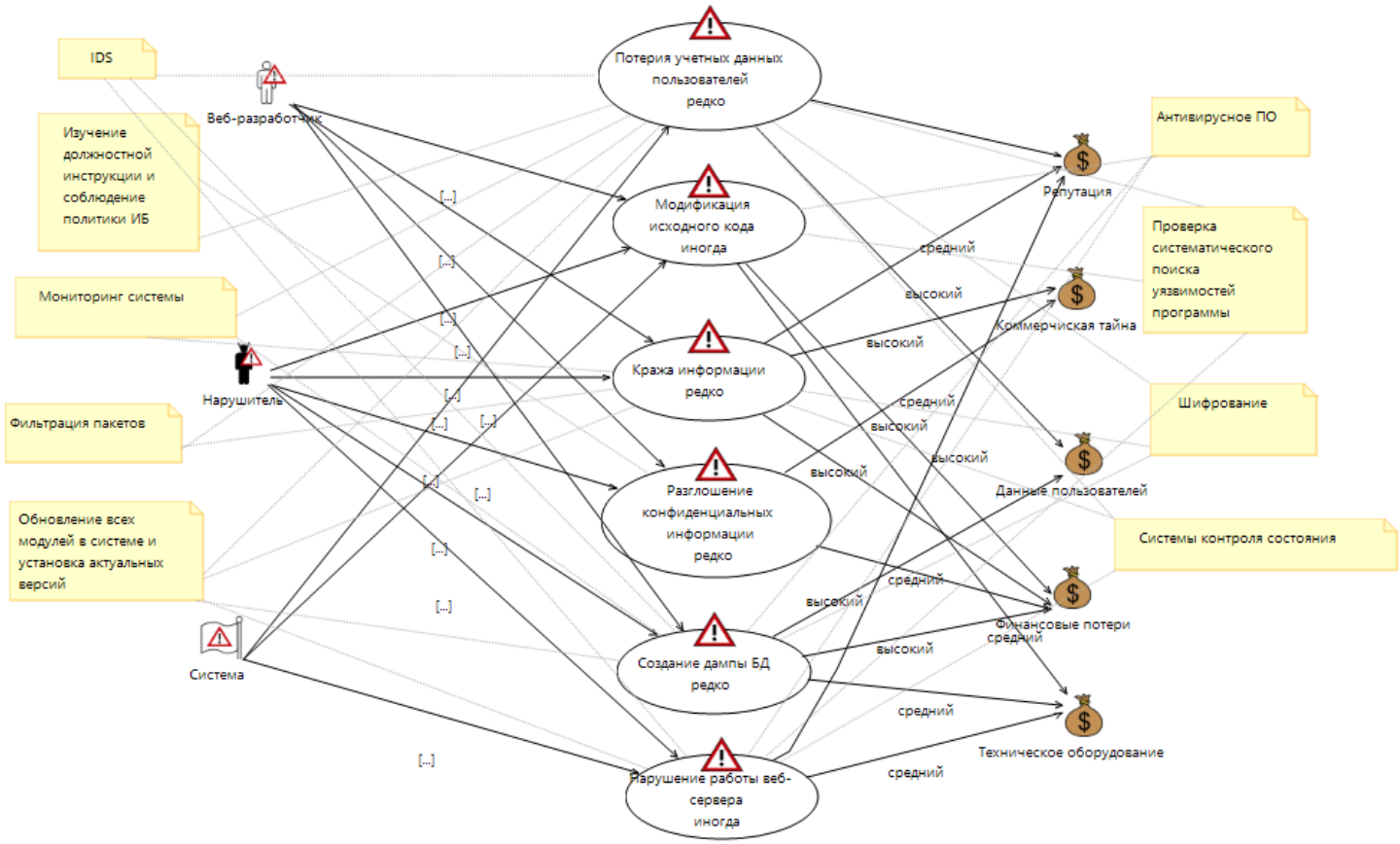
4.4 сурет – Тәуекелдер диаграммасы

Тәуекелдерді азайту барысында активтерімізге шараларды қолданамыз. Яғни, қорғау шаралары осалдықтар және қауіп-қатерлерді жүзеге асу тәсілдері арасында қосылды. 4.5 суреттегі әрбір осалдыққа қарсы іс-әрекеттер көрсетілген.



4.5 сурет – Қорғаныс элементтері бар қауіптер диаграммасы

Жоғарыда көрсетілгендей қорғау шараларын жүзеге асырғаннан кейінде қолайсыз тәуекелдер кездесуі мүмкін. Мұндай жағдайларда шешім қабылдаушылар қолайсыз тәуекелдердің жоғарғы қауіптерді бірінші кезекте жою керек. 4.6 суреттегі диаграммасында қолайсыз тәуекелдер сипатталады.



4.6 сурет – Қолайсыз тәуекелдер диаграммасы

4.3 Ақпараттық қауіпсіздік бөлімі бойынша қорытынды

Дипломдық жұмыстың осы бөлімінде біз веб-қосымшаның тәуекелдер сипаттамалары анықтадық. Ақпараттық қауіпсіздік тәуекелдерінің теориялық негіздері қарастырылды, екі параметрмен бағалау тәсілі көрсетілді,сондай-ақ ақпараттық қауіпсіздік тәуекелдерін бағалауды есептеу әдістемесі көрсетілді. Тәуекелдерді бағалау кезінде мыналар ескеріледі: ресурстардың құндылығы, қауіптер мен осалдықтардың маңыздылығы, қолданыстағы және жоспарланған қорғаныс құралдарының тиімділігі.

Келесі екінші бөлімінде CORAS құралымен тәуекелдерді талдау орындалды. Талдау барысында UML-диаграммалары қолданылып, тәуекелдердің қауіп диаграммасы және қорғау шаралары бойынша сызбалар сызылыды.

5 Өмір-тіршілік қауіпсіздігі бөлімі

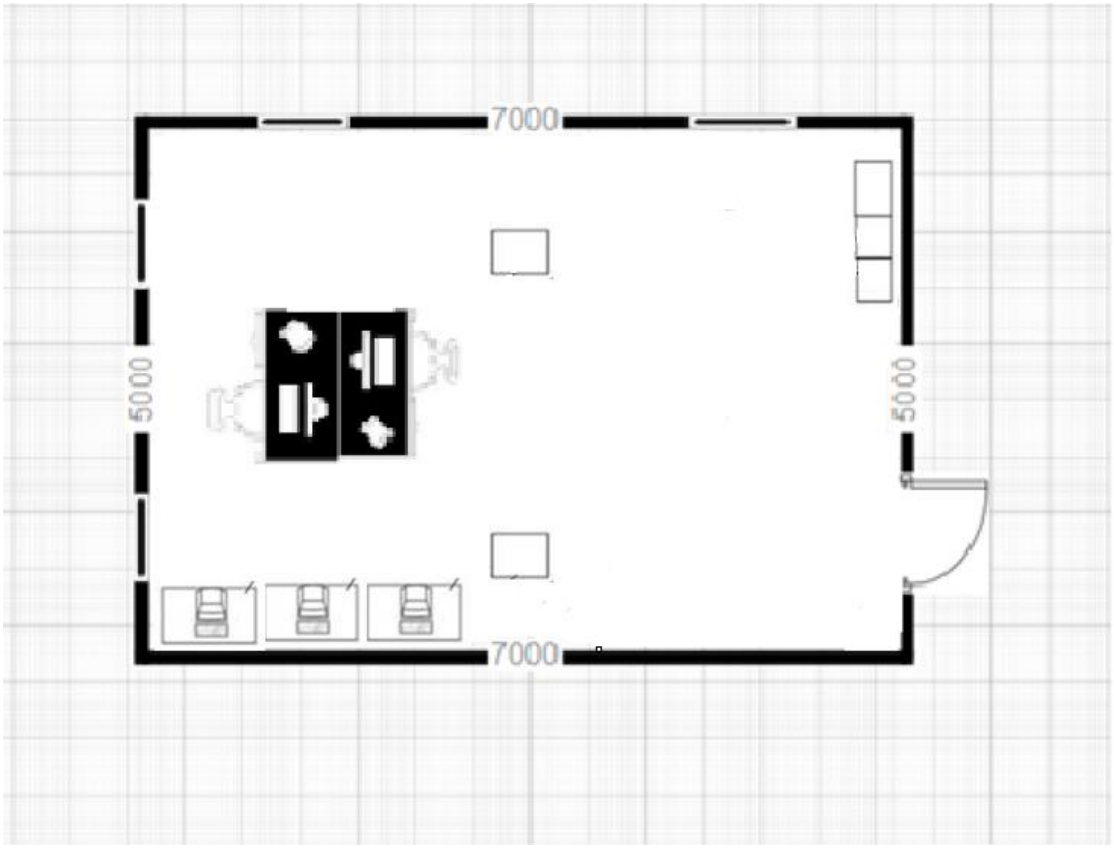
5.1 Жұмыс жағдайын талдау

Дипломдық жоба барысында «MD5 шифрлау әдісін қолдану арқылы қауіпсіз веб-қосымшаны әзірлеу» тақырыбы бойынша еңбек жағдайлары қарастырылды. Жұмыс орнында веб-бағдарламашының жеке компьютері, коммутатор және Ethernet сымы арқылы серверге қосылған құрылғы орналасқан. Жұмыс бөлмесінде ауа температурасы, ауа ылғалдылығын дұрыс сақталмағандықтан желдету жүйесі және алғашқы өрт сөндіру құралдары қажеттілігін есептеу керек болды. Сол себебті дипломдық жұмыстың бұл бөлімі алғашқы өрт сөндіру құралдарына қажеттілікті анықтау және желдету жүйесін есептеп, веб-бағдарламашының жұмыс істеуге ыңғайлы жағдай жасалу туралы шешім қабылданды.[16]

Бөлменің сипаттамасы:

- металлгалоген лампалар;
- сервер;
- компьютер;
- пластиктен жасалған жапқыштар;
- кондиционер;
- терезенің өлшемі 1,4 м*1,2 м;

5.1–суретте бөлменің параметрлері: ұзындығы 7м, ені 5м, биіктігі 2,6м;



5.1 сурет – Бөлмесінің сызбасы

5.1.1 Желдету жүйесі

Жұмыс орнын желдету кезінде бекітілген нормаларды қамтамасыз ету үшін ҚР ҚНжЕ 4.02-42-2006 [15] талаптарына сәйкес жасалынады. Адам ағзасына ауадағы метаболизм өнімдері, шаң және зиянды өндірістік заттар әсер еткендіктен желдету жүйесін қамтамасыздандыру тәсілдерін ұйымдастырады. Ауа алмасу оңтайлы шарттарына байланысты табиғи жолмен немесе желдеткіш құрылғылармен жүзеге асыру адамның жылу және функционалдық қалпына сәйкес әсер ету қолайлы жағдай жасалуына үлес қосады.

Желдету жүйесі бөлмедегі шаңмен ластанған ауаны және зиянды газдарды кетіруді қамтамасыз ететін ұйымдастырылған ауа алмасу, сондай-ақ микроклиматтық жағдайларды жақсарту ГОСТ 12.1.005-88 [16] талаптарына сәйкес қарастырылады.

Желдету жіктелу бойынша:

1) Ауа алмасуды ұйымдастыру әдісіне сәйкес - жалпы алмасу, ауаның ауысуы үй-жайдың толық көлемінде жүзеге асырылған кезде; жергілікті алмасу, онда бөлмедегі белгілі бір жерде әуе беріледі немесе жойылады.

2) Табиғи күштердің қозғалыс арқасында жасанды және ауа желдеткіштердің көмегімен қозғалады. Жел күші әрекеттесу әсерінен бөлме ауасы және сыртқы ауа салмағы қастығысқан жағдайда ауа алмасу механизмі туындайды.

3) Іштегі ауаны шығару немесе сырттан ауа үрлеу.

Жұмыс бөлмесінде серверге үлкен жүктеме түскендіктен қызып істен шығу қауіпі жоғарылайды, сондықтан желдету жылдамдығын арттыру туралы шешім қабылданады. Жұмыс бөлмесінде аэрациялық желдету қолдану негізінде ауа алмасу жылдамдығын 10-15 есеге арттырып, бөлме температурасы желдің бағытын саңыраулардың көмегімен ұйымдастырады. Табиғи желдету кезінде жылу мен жел қысымдары бір мезгілде әсер еткендіктен, сыртқы ауа жұмыс бөлмесінің төменгі тесіктерінен шығады. Желдің көлденең жағында орналасқан жел өткелі жабық болады. Ластанған ауаны жұмыс бөлмесіне кіргізбеуі үшін, ауа еденнен 3-3,5 метрден төмен емес орналасқан саңылауларды орналастыру арқылы шығарылады.

Егер табиғи желдеткіштердің тазалық талаптарына сай көрсеткіштерге жетпеген жағдайда механикалық желдетуді қолдану негізінде жүзеге асырылады. Қолайлы еңбек жағдайларын ұйымдастыру үшін SN 235-70 санитарлық нормаларына сәйкес механикалық желдетудің параметрлері сай орындалады:

– ауа температурасын, салыстырмалы ылғалдылықты қамтамасыз ету;

– климаттық жағдайларға тәуелсіз, талап етілетін көлемде жыл бойына біркелкі пайдалану;

– бөлменің кез келген нүктесінде ауаны жеткізу және ауа ағымынан бөлеу;

– құрылғының жергілікті сору қабілеті;

– бөлмеден алынатын желдетілетін ауаны тазарту.

Бөлменің желдету жүйесін қамтамасыздандыру барысында ескеретін заттар: 1 м^2 бөлме үшін 2 м^3 / сағ таза ауаны жеткізуді орнату.

ГОСТ 12.1.005-88 [16] санитарлық-гигиеналық талаптарына сай бөлмеде ауа ылғалдылығы және ауа жылдамдығына негізделе бір адамға сағатына ауа көлемі 60 м^3 құрайды.

5.1.2 Өрт қауіпсіздігі

Су қондырғыларына орнату ережесіне негізделе өртке қарсы ҚР ҚНжЕ 2.02-05-2009 [14] талаптары сәйкес жобаланады. Өрт туындаған жағдайда бөлме ішінде қымбат құрал-жабдықтардан бастап адам денсаулығына зардап тигізеді. Өңдеу барысында ақау туындау және құрал-жабдықтардың қолдану шарттарын ескермеу барысында өрттің пайда болуына себеб болады.

Жұмыс бөлмесінде өрт қауіпсіздігін жүзеге асыру үшін сақтану ережелерін ұстанады:

– электр құрылғылардың қолдану шарттарын қарау;

– ақаулы қалпында тұрған электр құрылғыларды ток көзіне салуға тыйым салынады;

– сымның нормаға сай қызу температурасын қадағалау;

– қосқыштарды, қуат сымдарын, жерге тұйықтау құрылғыларын, монитордың артқы жағына түртуге тыйым салынады;

– құрылғыларды өзіндік жөндеуге тыйым салу;

– электр лампаларының бетіне қағаз, шүберек және басқа да жанғыш

– материалдарды қоюға тыйым салынады;

– жоғарғы қуатты электр құрылғыларын бір розеткада қосуға болмайды.

Егер өрт туындаған жағдайда, қажет шаралар:

– дабыл батырмасын қосыңыз;

– бөлмедегі өрт сөндіргіш құралдарымен қолдану;

– өртті жою үшін сақтық шараларын қолданыңыз;

– мүмкіндігінше материалдық активтерді босату;

– тез жанатын құрал-жабдықтардан қорғану;

– ток көздеріне қосылған құрылғыларды өшіріңіз;

– тиісті қызметтерге өрт туралы есеп беру – кезекші, басқарушы

– бақылау пункті.

Өрттің туындаған жағдайда бөлмеде жылу тез қабылдайтын заттардан арақашықтық сақтап, барлық электр құрылғыларын ток көздерінен ажыратыңыз.

5.2 Есептеу бөлімі

5.2.1 Бөлменің желдету жүйесін есептеу

Желдету жүйесі арқылы бөлмедегі ауа температурасын, салыстырмалы ылғалдылығын, қозғалыс жылдамдығын және ауа қысымын баптауға қол жеткізеді. Есептеу ҚНжЕ 4.02-42-2006 [16] бойынша орындалады.

5.1 формуласы бойынша артық жылуды анықталады:

$$Q_{АРТ} = Q_{өж} + Q_{жжж} + Q_{К} + Q_{Р} + Q_{тжж}, \quad (5.1)$$

мұндағы $Q_{өж}$, $Q_{жжж}$, $Q_{К}$ – өндірістік жабдықтардың көп санымен, жасанды жарықтандыру жүйесімен және жұмыс істейтін қызметкерлерден (адамдардан) бөлінетін жылу (ккал/сағ);

$Q_{Р}$ – күн сәулесімен енгізілетін жылу (күн радиациясы), (ккал/сағ);

$Q_{тжж}$ – табиғи жолмен жылу беру, (ккал/сағ).

Өндірістік жабдықпен бөлінетін жылуды 5.2 формуласы арқылы анықталады:

$$Q_{өж} = 840 * P_{ОБ} * \eta, \quad (5.2)$$

мұндағы 840 – жылу эквиваленті 1 (кВт/сағ);

$P_{ОБ}$ – жабдық тұтынатын қуат, (кВт/сағ);

η – жылудың бөлмеге өту коэффициенті.

Жарық беретін қондырғылардан бөлінетін жылуды 5.3 формула арқылы анықталады:

$$Q_{жжж} = 1000 * N, \quad (5.3)$$

мұндағы N – шамдардың шығыс қуаты.

$$Q_{жжж} = 1000 * 0{,}22 = 220 \text{ ккал/сағ},.$$

Қызметкерлерден бөлетін жылуды 5.4 формуласы арқылы анықталады:

$$Q_{К} = К_{С} * (q - q_{БУ}), \quad (5.4)$$

мұнда $К_{С}$ – жұмыс істейтін кызметкерлер саны;

$(q\text{-}q_{БУ})$ – анық жылу, (ккал/сағ);

q – I-III жұмыс санаты кезінде бір адамнан жылу бөлінуі, (ккал/сағ);

$q_{БУ}$ – жылудың буланууына жұмсалған жылу (ккал/сағ).

Бөлмедегі жұмыс II категорияға жатады, $Q_{К}$ есептеп шығарамыз

$$q=220 \text{ ккал/сағ},$$

$$q_{БУ}=120 \text{ ккал/сағ},$$

$$Q_{К} = 10 * (220 - 120) = 1000 \text{ккал/сағ}.$$

Күн радиациясымен енгізілетін жылуды 5.5 формуласы арқылы анықталады:

$$Q_{Р} = m * F * q_{ОСТ}, \quad (5.5)$$

мұнда m – бөлмедегі терезелер саны;

F – бір терезенің ауданы ($м^2$);

$q_{Ш}$ – шыныланған терезе арқыла күн радиацисы (беті, яғни ауданы 1 ($м^2$) шыныланған бет арқылы бір сағат ішінде енгізілген жылу мөлшері).

Екі әйнектелген ағаш түптелген терезе үшін $q_{Ш}$ =125 (терезелер оңтүстік-шығысқа шығады). Терезелер саны 1 тең.

$$Q_Р = 4 * 4 * 125 = 2000 \text{ ккал/сағ.}$$

Жылдың жылы кезеңі үшін есептеу кезінде $Q_{ӨЖ} = 0$ деп есептеуге болады.

$$Q_{АРТ} = 2042{,}5 + 220 + 1000 + 2000 = 5262{,}5 \text{ккал/сағ.}$$

Жылу артық болған жағдайда бөлмеден шығару қажет ауа мөлшері 5.6 формула бойынша болады:

$$L_b = \frac{Q_{АРТ}}{C_b * \Delta t g \gamma_b}, \quad (5.6)$$

мұнда $Q_{АРТ}$ – артық жылу, (ккал/сағ);

C_b – ауаның жылу сыйымдылығы (0,24 ккал/кг°С);

γ_b=1,206 (кг/куб.м.) - ауаның салыстырмалы салмағы.

5.7 формуласының көмегімен Δt температурасының айырмашылығы анықталады:

$$\Delta t = t_{шығ} - t_{түс}, \quad (5.7)$$

мұнда $t_{шығ}$ – бөлмеден шығатын ауаның температурасы,°С;

$t_{түс}$ – бөлмеге түсетін ауаның температурасы, °С;

Егер ауаның жылу сыйымдылығы Q < 20 Н (ккал/м^3) болса, онда Δt = 4 (°С), ал Q > 20 Н (ккал/м^3), Δt = 8 (°С) қабылданады.

Есептеу кезінде Δt шамасы 5.8 формуласынан ауаның жылу кернеулігіне байланысты таңдалады:

$$Q_Н = \frac{Q_{АРТ}}{V_H}, \quad (5.8)$$

$$L_b = \frac{5262{,}5}{0{,}24 \times 4 \times 1{,}206} = 1736.64 \text{ м}^3/\text{сағ.}$$

Ауа шығыны 1620 м^3/ч, жоғары жіберумен сплит-жүйелі кондиционері тандалады. 5.9 формуласынан кондиционерлер санын анықталады:

$$N = \frac{L_b}{1620}, \quad (5.9)$$

мұнда N – кондиционердің саны.

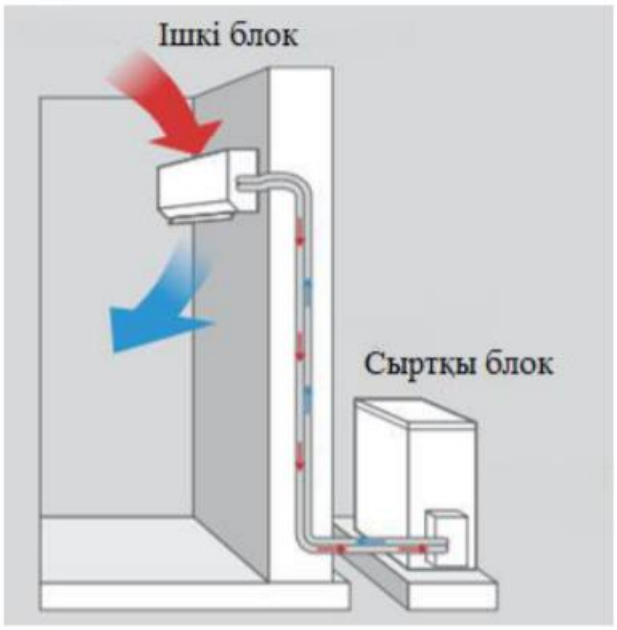
$$N = \frac{1736.64}{1620} \approx 1 \text{ кондиционер.}$$

Зерттелген деректерге сүйене отырып, жұмыс бөлмесіне SNOWCAP-09BB-1 кабырғаға орнатылатын тұрмыстық кондиционері таңдалады. Ол бізге қажетті бөлменің ауа саны бойынша алынған мәліметтерді толықтай қанағаттандырады. Кондиционер арқылы бөлмедегі ауа температурасын, салыстырмалы ылғалдылығын, қозғалыс жылдамдығын және ауа қысымын баптауға қол жеткізеді. Бөлмедегі кондициялау қондырғылары ауаны бактериологиялық тазарту, жағымсыз иістерді залалсыздандыру, оттек құрамын байыту жүзеге асырады.

SNOWCAP-09BB-1 кондиционерінің жалпы сипаттамалары төмендегі кестеде көрсетілген.

5.2 - кесте – Кондиционердің негізгі техникалық сипаттамалары

Ішкі блоктың өлшемдері	Қуаты, Вт		Алаң, $м^2$	
720x200x270	Салқындату режиміндегі қуат	Қыздыру режиміндегі қуат	Жылыту алаңы	Салқындату алаңы
	2650	2700	40	40



5.2 сурет – Кондиционердің ауа алмасу үдерісі

5.2.2 Өрт қауіпсіздігін есептеу

Өрт қауіпсіздігі бөлмедегі қызметкерлердің қолайлы жағдайын қамтамасыз етеді. Есептеу барысында ҚНжЕ 2.02-05-2009 [15] әдістемелік нұсқауымен орындалды. Кәсіпорында өрт жану үдерісін оқшаулау немесе жою үшін алғашқы өрт сөндіру құралдары пайдаланылады. Мысалы, сынық, балта, әмбебап ілгек құралдары жанып жатқан конструкцияларды бұзуға, ‘эвакуациялау’ жолдарын ашу үшін пайдаланылады. Багор өрт сөндіру құралы арқылы жанып жатқан орта заттарды алып тастау үшін қолданылады. Өрттің аз аймағын ауа кіру арқылы оқшаулау мақсатында асбест және брезент жапқыштар қолданылады. Шелектер сумен өрт сөндіру үшін қолданылады. Күректер ошақты құммен, жермен жабу үшін қолданылады.

Кәсіпорында өрт сөндіргіштер мен өрт құрал-саймандары (сынықтар, багрлер, балталар, шелектер, жапқыштар) орналастырылатын өрт қалқандары орнатылады. Қалқанның жанында құм мен күрегі бар жәшік, сондай-ақ көлемі 200-250 л су бар бөшке орнатылады.:

– егер кәсіпорын ішкі өртке қарсы су құбырымен және автоматты өрт сөндіру қондырғыларымен жабдықталмаған жағдайда;

– егер кәсіпорын аумағында сыртқы өртке қарсы су құбыры болмаса;

– егер сыртқы өрт су көздері ғимараттан, кәсіпорындардың сыртқы технологиялық қондырғыларынан 100 м артық қашықтыққа алыстатылған болса.[15]

ҚНжЕ 2.02-05-2009 [14] сәйкес, ғимарат өрттің даму қауіптілік дәрежесіне, жанғыш материалдардың функционалдық мақсатына және өрт жүктемесіне байланысты l-санаттағы топқа жатады. Өрт себептері:

– кернеудің нормадан тыс әсер етуі;
– сымдардың ақау әсері;
– сулы немесе құрғақтық арқылы электр құрылғыларымен жұмыс істеу;
– жоғары қуат көздерін әлсіз токка қосу;
– жанғыш заттардың электр құралдарымен әрекеттесу.

Бөлмеде оттың пайда болу көздеріне: ақау нәтижесінде қызып кететін құрылғылар, құрамында жану элементтері бар материалдар, техникалық қызмет көрсету үшін пайдаланылатын құрылғылар, электр қондырғылары және т.б. [14].

Өрт сөндіру құралы ретінде көмірқышқыл газы-хладонның аралас қоспасы қолданылады. Көлемді өрт сөндіру үшін md көміртек-хладон қос тотығының біріктірілген композициясының есептелген салмағы 5.10 формуласы шамасымен анықталады:

$$\mathrm{md} = \mathrm{k} \cdot \mathrm{gn} \cdot \mathrm{V}, \qquad (5.10)$$

мұнда k=1,2 – көмірқышқыл хладон құрамының ескерілмейтін шығындарын өтеу коэффициенті;

gn=0,04 көмірқышқыл-хладон құрамының нормативтік массалық концентрациясы.

V – бөлме көлемі 5.11 формуласы бойынша анықталады:

$$V = A \cdot \mathrm{B} \cdot \mathrm{H}, \qquad (5.11)$$

мұндағы, А = 7м – бөлменің ұзындығы;
В = 5м — бөлменің ені;
Н = 2,6м — бөлменің биіктігі.

Сонда:

х баллондарының есептік саны 12 литрлік 9.5 кг көмірқышқыл-хладон құрамының сыйымдылығы есебінен анықталады.

Магистральдік құбырдың ішкі диаметрі di(мм) 5.12 формуласы бойынша анықталады:

$$di = 12\ 32\ = 17\text{мм}, \qquad (5.12)$$

12 магистральдік құбырдың эквивалентті ұзындығы 5.13 формула бойынша анықталады:

$$12 = k1 \cdot 1, \qquad (5.13)$$

мұнда k1=1,2 – жергілікті ысыраптарды ескермейтін өтем үшін құбыр ұзындығының ұлғаю коэффициенті;

l=2,6м – жоба бойынша құбырдың ұзындығы, сонда:

$$12 = 1{,}2 \cdot 2{,}6 = 3{,}12 \text{ м.}$$

Құбырдың эквивалентті ұзындығы мен диаметріне байланысты Q көмірқышқыл-хладон құрамының шығыны 1,4 кг/с тең.

Көмірқышқыл-хладон құрамын берудің есептік уақыты t, 5.14 формуласы бойынша анықталады:

$$t = \frac{md}{60 \cdot Q}, \qquad (5.14)$$

Сонда,

$$\text{T} = \frac{4{,}368}{128 \cdot 1{,}4} = 0{,}0243\text{мин.}$$

Көмірқышқыл-хладон құрамының негізгі қорының массасы 5.15 формуласы бойынша анықталады:

$$\text{M} = 1{,}1 \cdot \text{md} \cdot (1 + \text{k2} \cdot \text{k1}) \qquad (5.15)$$

мұндағы к2 = 0,2-баллондар мен құбырлардағы көмірқышқыл-хладон құрамының қалдығын ескеретін коэффициент.

Сонда:

$$\text{M} = 1{,}1 \cdot \text{md} \cdot (1 + 0{,}2/1{,}2) = 5{,}605 \text{ кг.}$$

Осылайша, алынған нәтижелерден автоматты өрт сөндіру жүйесінің қалыпты жұмыс істеуін қамтамасыз ету үшін сыйымдылығы 12 литр көмірқышқыл-хладон құрамының 1 баллоны қажет, қоспаның салмағы 9.5 кг. Автоматты газды сөндіру қондырғыларында автоматты іске қосуға арналған құрылғылар бар.

5.3 Өмір-тіршілік қауіпсіздігі бөлімі бойынша қорытынды

Бөлім бойынша қорытынды: Қызметкерлердің еңбек жағдайына талдау жүргізілді, талдау барысында зиянды факторлар анықталды. Бөлмедегі желдету жүйесі және өрт қауіпсіздігі туралы мәселе талқыланып есептеулер жүргізілді. Жүргізілген есептеулерге сүйеніп, бөлмедегі қолайлы жағдай көбінесе ауа ағынының дұрыс бөлінуіне байланысты екені анықталды, сол себепті SNOWCAP-09BB-1 кондиционерін бөлменің терезе жанына орнатуға шешім

қабылданды. Сондай-ақ қолайлы еңбек жағдайларын жасау үшін бөлмедегі өрт қауіпсіздік есептеуі жүргізілді.

Қорытынды

Қорытындылай келе, осы дипломдық жобада қолданушы енгізген жеке деректерін қауіпсіз түрде сақтайтын веб-қосымша әзірленді және веб-қосымшаға төнетін қауіптерден қорғану жолдары қарастырылды.

Дипломдық жоба бойынша веб-қосымшаның құру кезеңдері қарастырылып, негізгі алты кезеңдердің ішінен веб-қосымшаны құрудың тиімді тәсілі таңдалды. Сонымен қатар қауіпсіздік шаралары талқыланды. Қауіпсіздік шаралары қарапайым веб-қосымшаларға HTTPS хаттамасын қолдану, бағдарламалық қамтамасыз ету жүйелерін уақылы жаңарту, ақпаратты сұрыптау қосымшаларын қолдану және құпия мәліметтерді шифрлау және оны тексеру туралы ұсыныстары көрсетілді.

Веб-қосымшаның қолданушымен әрекеттесу алгоритмі құрастырылып, қолданушыға тиімді және қарапайым тексеру кезеңдері жасалды. Тексеріс кезінде қолданушы енгізген ақпарат жергілікті серверде өңдеу үдерісі арқылы орындалды. Әзірлеу барысында клиенттік және серверлік технологияларымен веб-қосымшаның интерфейсі құрылды. Веб-қосымшаның интерфейсі қолданушы үшін ыңғайлы прототипі клиенттік құралдардың көмегімен жасалынды. Сондай-ақ веб-қосымшаның прототипі бірнеше элементтерді қамтыды:

- басты парақша;
- авторизация парақшасы;
- тіркелу парақшасы;
- тауарлар бөлімі;
- таңдалған тауады төлем парақшасы.

Веб-қосымшаны құруда HTML5, CSS3, JavaScript, PHP, MySQL, OpenServer технологиялары қолданылды. HTML5, CSS3, JavaScript бағдарлалық тілдерін клиент жағында интерфейсі жасалынды, ал PHP серверлік бағдарламалау тілі арқылы қорғаныс механизмін іске қосылды. MySQL деректер қоры және жергілікті OpenServer сервердің арасындағы байланыс орнатылды. Сондай-ақ қолданушының жеке ақпаратың қорғау, яғни парольдік қорғау механизмі жүзеге асырылып, деректер қорында сақталынды. Парольдік қорғау механизмі MD5 шифрлау алгоритмі арқылы сенімді қорғанысты қамтамасыздандырылды. Бұл алгоритм еркін ұзындықтағы хабарлама дайджестерін жасап, 128 биттік хеширлеу арқылы жүзеге асырылды.

Ақпараттық қауіпсіздік тәуекелдер бөлімінде веб-қосымшаның тәуекелдер сипаттамалары анықталып, екі параметрмен бағалау тәсілі көрсетілді.

Өмір-тіршілік қауіпсіздік бөлімінде желдету жүйесі және өрт қауіпсіздігі туралы мәселе талқыланып есептеулер жүргізілді. Жүргізілген есептеулерге сүйеніп, бөлмедегі қолайлы жағдай көбінесе ауа ағынының дұрыс бөлінуіне байланысты екені анықталды, сондай-ақ бөлмедегі өрт қауіпсіздік есептеуі жүргізілді.

Әдебиеттер тізімі

1 Пьюривал С. Основы разработки веб-приложений. – СПб. : Питер, 2015. – 272 с.

2 Книга веб-программиста. Секреты профессиональной разработки веб-сайтов / Б.Хоган и др. - Москва: Мир, 2013. - 288 с.

3 Мэтью, Дэвид HTML5. Разработка веб-приложений / Дэвид Мэтью. - М.: Рид Групп, 2012. - 320 с.

4 Фиртман, Максимилиано jQuery Mobile. Разработка приложений для смартфонов и планшетов / Максимилиано Фиртман. - М.: БХВ-Петербург, 2013. - 256 с.

5 Взломать пароль — по-прежнему легко / (http://hi-tech.mail.ru/news/item/2932/).

6 Балабанов И.Т. Торговля через виртуальный магазин [Текст] / И. Т. Балабанов //Электронная коммерция, 2004. – С. 195-197.

7 Секреты хакеров. Безопасность Web-приложений – готовые рещения. :Пер.с англ. – М.: Издательский дом “Вильямс”,2003. – 384 с.:ил – Парал.Тит.англ.

8 Крис Митчелл. Артем Конев. Обеспечение безопасности веб-сайтов. // Australia: SophosLabs. [Электронный ресурс]. URL: https://help.yandex.ru/webmaster/protecting-sites/contents.xml (15.02.2019).

9 Бирюков А.А. Информационная безопасность: защита и нападение – М.:ДМК Пресс, 2012.-474.: ил.

10 Are Your Web Apps Protected Against Component Vulnerabilities? // Защищены ли компоненты используемые в ваших веб приложениях от уязвимостей? [Электронный ресурс]. 2019. URL https://www.tenable.com/blog/are-your-web-appsprotected-against-component-vulnerabilities/ (22.04.2019).

11 https://webcase.com.ua/blog/bezopasnost-web-prilozhenij/#id2.

12 Грекул В. И. Проектное управление в сфере информационных технологий / В. И. Грекул, Н. Л. Коровкина, Ю. В. Куприянов. – М. : Бином. Лаборатория знаний, 2013. – 336 с.

13 Жандаулетова, Ф. Р. Охрана труда: учебник для вузов / Ф.Р. Жандаулетова, Т.Е. Хакимжанов, Т.С. Санатова; МОН РК, НАО АУЭС. - Алматы : АУЭС, 2019. - 399 с.

14 ҚР ҚНжЕ 4.02-42-2006. «Жылыту, желдету және ауа баптау» - Сәулет, қала құрылысы және құрылыс саласындағы мемлекеттік нормативтер, Астана, 2007.

15 Абикенова А.А., Санатова Т.С. Безопасность жизнедеятельности. Методические указания к выполнению раздела «Пожарная профилактика» в выпускных работах для всех специальностей. Бакалавриат - Алматы: АИЭС, 2009. - 32 с.

А қосымшасы
(міндетті)

Программа листингі

```
<?php
require "db.php";
?>
!
<!DOCTYPE html>
<html lang="kz">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="css/bootstrap-grid.min.css">
<link rel="stylesheet" href="css/style.css">
<title>Веб</title>
</head>
<body>
<div class="main">
<div class="container">
<div class="main-block">
<h1 class="main-title">
Нет более искренней любви, <span>чем любовь к еде.</span>
</h1>
<!-- <span class="main-span">Чтобы заказать нужно авторицоваться</span> -->
<?php if(isset($_SESSION['logged_user']) ) : ?>
<span style="font-size: 25px;">Поздравляю, <?php echo $_SESSION['logged_user']->name; ?>!</span><br>
<span style=" font-size: 30px;">Вы успешно авторизовались</span><br>
<hr>
<button class="button"><a style="text-decoration: none;" href="/logout.php">Выйти</a></button>
<?php else : ?>
<button class="main-button button"><a href="/login.php">Авторизация</a></button>
<button class="main-button--two button"><a href="/signup.php">Регистрация</a></button>
<?php endif; ?>

</div>

</div>
</div>
```

```
<div class="menu-nur">
<div class="container">
<div class="section-title">Меню</div>
<div class="menu-nur-block">
<div class="menu-nur__item">
<div class="menu-nur__img">
<img src="img/menu-nur-img1.png" alt="menu-nur">
</div>
<div class="menu-nur__title">ОСНОВНЫЕ БЛЮДА</div>
<button class="button menu-nur__get"><a
href="oplata.php">Купить</a></button>
</div>
<div class="menu-nur__item">
<div class="menu-nur__img">
<img src="img/menu-nur-img2.png" alt="menu-nur">
</div>
<div class="menu-nur__title">СУПЫ</div>
<button class="button menu-nur__get"><a
href="oplata.php">Купить</a></button>
</div>
<div class="menu-nur__item">
<div class="menu-nur__img">
<img src="img/menu-nur-img3.png" alt="menu-nur">
</div>
<div class="menu-nur__title">ГАМБУРГЕРЫ</div>
<button class="button menu-nur__get"><a
href="oplata.php">Купить</a></button>
</div>
</div>

<div class="menu-nur-block">
<div class="menu-nur__item">
<div class="menu-nur__img">
<img src="img/menu-nur-img4.png" alt="menu-nur">
</div>
<div class="menu-nur__title">ДЕСЕРТЫ</div>
<button class="button menu-nur__get" ><a
href="oplata.php">Купить</a></button>
</div>
<div class="menu-nur__item">
<div class="menu-nur__img">
<img src="img/menu-nur-img5.png" alt="menu-nur">
</div>
<div class="menu-nur__title">БАРБЕКЮ</div>
```

```
<button class="button menu-nur__get" ><a href="oplata.php">Купить</a></button>
</div>
<div class="menu-nur__item ">
<div class="menu-nur__img">
<img src="img/menu-nur-img6.png" alt="menu-nur">
</div>
<div class="menu-nur__title">САЛАТЫ</div>
<button class="button menu-nur__get"><a href="oplata.php">Купить</a></button>
</div>
</div>
</div>
</div>
<div class="app">
<div class="container">
<h2 class="app-text">Приложения показывается адаптивна на всех <span>мобильных устройствах</span></h2>
</div>
</div>
<script src="https://code.jquery.com/jquery-1.12.4.min.js" integrity="sha256-ZosEbRLbNQzLpnKIkEdrPv7lOy9C27hHQ+Xp8a4MxAQ=" crossorigin="anonymous"></script>
</body>
</html>
<?php
require "db.php";

$data = $_POST;
if( isset($data['do_login']) )
{
$errors = array();
$user = R::findOne('users', 'login = ?', array($data['login']));
if( $user)
{
//логин существует
if(password_verify($data['password'], $user->password)){
$_SESSION['logged_user'] = $user;
echo '<div class="message">Вы успешно авторизовались!<br/>
Можете перейти на <a href="logsuc.php">главную</a> страницу!</div>';
}else
{
$errors[] = 'Неверно введён пароль!';
```

```
}
}else
{
$errors[] = 'Пользователь с таким логином не найден!';
}

if(! empty($errors) )
{
echo '<div class="message error">'.array_shift($errors).'</div>';
}
}
?>
<!DOCTYPE html>
<html lang="kz">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>LOGIN</title>
<link rel="stylesheet" href="css/bootstrap-grid.min.css">
<link rel="stylesheet" href="css/style.css">
</head>
<body>
<div class="login-block main-content dark-bg">
<div class="container">
<form action="login.php" method="POST" class="global-form">
<input name="login" value="<?php echo @$data['login'];?>" type="text" placeholder="Ваш Логин:" class="global-input"><br>

<input name="password" value="<?php echo @$data['password'];?>" type="password"  placeholder="Ваш пароль:" class="global-input"><br>

<button type="submit" name="do_login" class="global-button button ">Войти</button>
</form>
</div>
</div>
</body>
</html>
```