

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
им. ГУМАРБЕКА ДАУКЕЕВА»
Кафедра IT – инжиниринг

«ДОПУЩЕН К ЗАЩИТЕ»
Зав. кафедрой PhD, доцент Досжанова А.А.
_____ « ____ » _____ 2020 г.

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка системы «Электронное общежитие» для высшего учебного заведения

Специальность 5В070400 – Вычислительная техника и программное обеспечение

Выполнила: Гумарбекова К.М Группа ВТ-16-2

Научный руководитель: PhD, доцент Досжанова А.А

Консультанты:

по экономической части: к.э.н., профессор Габелашвили К.Р

(учёная степень, звание, Ф.И.О.)

_____ « ____ » _____ 2020 г.

по безопасности жизнедеятельности: доцент Приходько Н.Г

(учёная степень, звание, Ф.И.О.)

_____ « ____ » _____ 2020 г.

по программному обеспечению: ст. преп. Майкотов М.Н

(учёная степень, звание, Ф.И.О.)

_____ « ____ » _____ 2020 г.

Нормоконтролер: ст. преп. Абсатарова Б.Р

(учёная степень, звание, Ф.И.О.)

_____ « ____ » _____ 2020 г.

Рецензент: _____

(учёная степень, звание, Ф.И.О.)

_____ « ____ » _____ 2020 г.

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
им. ГУМАРБЕКА ДАУКЕЕВА»

Институт систем управления и информационных технологий

Кафедра IT-инжиниринг

Специальность 5В070400 – «Вычислительная техника и программное обеспечение»

ЗАДАНИЕ

на выполнение дипломного проекта

Студенту Тумарбековой Камилле Максutowне

Тема проекта: Разработка системы «Электронное общежитие» для высшего учебного заведения

Утверждена приказом по университету № ___ от «___» _____ 2020 г.

Срок сдачи законченного проекта «___» _____ 2020 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): руководство по программированию на PHP, интегрированная среда Plesk, данные преддипломной практики.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта:

- а) анализ и исследование предметной области;
- б) проектирование приложения;
- в) реализация приложения;
- г) экономическое обоснование эффективности;
- д) вопросы безопасности жизнедеятельности и охраны труда.

Перечень графического материала (с точным указанием обязательных чертежей): имеется 18 таблиц, 62 иллюстрации и 1 диаграмма.

Основная рекомендуемая литература:

1 Никсон, Робин Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript и CSS / Робин Никсон. - М.: "Издательство "Питер", 2013. - 560 с.

2 Прохоренок, Николай HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера (+ CD-ROM) / Николай Прохоренок. - М.: БХВ-Петербург, 2012. - 912 с.

3 Дронов В. PHP, MySQL и Dreamweaver. Разработка интерактивных Web-сайтов; БХВ-Петербург - М., 2016. - 480 с.

Консультация по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Подготовка экономической части дипломного проекта	Габелашвили К.Р	24.04.2020	
Подготовка части БЖД	Приходько Н.Г	24.04.2020	
Программная часть	Майкотов М.Н	14.05.2020	
Нормоконтроль	Абсатарова Б.Р	18.05.2020	

ГРАФИК
подготовки дипломной работы (проекта)

Наименования разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечания
Анализ и исследование предметной области	01.11.2019- 20.12.2019	
Проектирование приложения	21.12.2019- 20.02.2020	
Программная реализация	21.02.2020 20.04.2020	

Дата выдачи задания « ____ » _____ 2020 г.

Заведующий кафедрой _____ А.А.Досжанова
(подпись)

Научный руководитель проекта _____ А.А.Досжанова
(подпись)

Задание принял к исполнению студент _____
М.К.Тумарбекова
(подпись)

АНДАТПА

Дипломдық жобаның тақырыбы: «Жоғары оқу орны үшін «Электрондық жатақхана» жүйесін дамыту».

Жоғары оқу орны үшін әзірленген «Электрондық жатақхана» жүйесі жатақхана коменданттарының және студенттер жатақханаларын тексеруге жауапты барлық қызметкерлердің жұмысын автоматтандыруға арналған. Сондай-ақ бұл жүйе сыбайлас жемқорлық факторын жоя отырып, студенттердің қоныстануында ашықтық пен әділеттілікті қамтамасыз етеді.

Дипломдық жоба web - платформа форматында орындалды. HTML, CSS, PHP, JavaScript сияқты технологиялар әзірленген дипломдық жобаның мақсатына жету үшін пайдаланылады.

Сонымен қатар, бағдарламаны әзірлеудің шығындары мен бағасының қаржылық есебі, әзірленетін жобаның қаржылық қажеттілігін бағалау жүргізіледі және өнеркәсіптік жарықтандырудың сапасын арттыру бойынша іс-шаралар ұсынылады.

АННОТАЦИЯ

Тема дипломного проекта: «Разработка системы «Электронное общежитие» для высшего учебного заведения».

Разработанная система «Электронное общежитие» для высшего учебного заведения предназначена для цифровизации работы комендантов общежитий и всех сотрудников, отвечающих за заселение в студенческие общежития. А также данная система демонстрирует прозрачность при заселении студентов.

Дипломный проект был выполнен в формате web - платформы. Такие технологии, как HTML, CSS, PHP, JavaScript были использованы для достижения цели разрабатываемого дипломного проекта.

Кроме того, был проведен финансовый расчет издержек и цены разработки программы, оценка финансовой необходимости разрабатываемого проекта и были предложены мероприятия по повышению качества промышленного освещения.

ABSTRACT

Theme of the graduation project: "Development of the system "Electronic Dormitory" for a higher educational institution".

The developed "Electronic Dormitory" system for a higher educational institution is designed to automate the work of the commandants of dormitories and all employees responsible for checking into student dormitories. And also this system provides transparency and justice in the settlement of students, eradicating the corruption factor.

The diploma project was made in the format of a web platform. Technologies such as HTML, CSS, PHP, and JavaScript were used to achieve the goal of the diploma project being developed.

In addition, a financial calculation of the cost and price of developing the program, an assessment of the financial need for the project being developed, and measures to improve the quality of industrial lighting were proposed.

Содержание

Введение.....	8
1 Обзор и анализ предметной области.....	10
1.1 Эффективность внедрения информационных систем.....	10
1.2 Анализ существующих аналогов систем.....	10
1.3 Постановка задачи.....	11
2 Проектирование системы и обоснование выбора инструментов.....	13
2.1 Проектирование системы принятия заявлений.....	13
2.2 Выбор инструментов для разработки платформы.....	16
3 Разработка программного продукта.....	21
3.1 Системные требования.....	21
3.2 Описание разработанных функций.....	21
3.3 Описание клиентской части.....	24
4 Техничко-экономическое обоснование.....	33
4.1 Определение сложности разработки ПО.....	33
4.2 Расчет затрат на разработку ПО.....	34
4.3 Расчет затрат на электроэнергию.....	36
4.4 Расчет затрат на оплату труда.....	37
4.5 Расчет затрат по социальному налогу.....	39
4.6 Амортизация основных фондов и прочие затраты.....	39
4.7 Определение возможной (договорной) цены ПО.....	42
4.8 Сравнительный анализ эксплуатационных затрат до и после внедрения.....	43
5 Безопасность жизнедеятельности.....	48
5.1 Анализ потенциально опасных и вредных факторов в офисе, воздействующих на персонал.....	48
5.2 Расчет искусственного освещения.....	49
5.3 Расчет систем кондиционирования рабочего помещения ИТ отдела.....	51
Заключение.....	55
Список литературы.....	56
Приложение А Техническое задание.....	58
Приложение Б Листинг программы.....	60

Введение

Актуальность темы заключается в том, что согласно статистике МОН РК на 2017-2018 учебный год, в вузах страны обучалось 496,2 тысяч студентов, из которых 40%, или около 203 тысяч, являлись иногородними, то есть потенциально нуждающимися в решении жилищного вопроса на период обучения. По последним данным, острый дефицит мест испытывают 17,4 тысяч студентов. В разрезе регионов наибольшая нехватка студенческого жилья, или 56%, приходится на города Нур-Султан и Алматы. В рамках реализации третьей из пяти социальных инициатив Главы государства в части улучшения условий проживания студенческой молодёжи необходимо до конца 2022 года обеспечить студентов ВУЗов и колледжей свыше 75 тысяч койко-мест. На строительство общежитий государство планирует выделить 118 млрд тенге в течение десяти лет.

И в связи с таким спросом в студенческих общежитиях внедрение системы электронного принятия является необходимостью, так как решение о заселении принимает определенный круг лиц, который имеет прямой контакт с заявителем (студентом), а также данные о заявлениях, о самих студентах не отцифровываются. В СМИ уже приводятся примеры, что в высших учебных заведениях уже были случаи, когда коменданты сдавали комнаты посторонним людям. А некоторые гостеприимные студенты приводили в общежитие своих друзей и надолго.

Многие образовательные учреждения имеют изъяны в автоматизации и цифровизации внутренних процессов, будь то налаживание дистанционного обучения, проведения экзаменов, принятие документов на поступление в университет или на военную кафедру, а также в дальнейшем это отразится и на подаче заявлений на заселение в общежитие. Современная IT-платформа, позволяющая едино принимать и управлять всеми полученными заявлениями и документами – это то, что поможет не просто держаться на плаву, но и обеспечить потенциал для снижения расходов и ускорения процессов работы.

В наши дни можно с уверенностью говорить о том, что цифровизация – это необходимость 21 века и среди университетов есть полное понимание, на чем можно сэкономить, а без чего невозможно организовать стабильную работу.

Профессиональная система — это надежный инструмент, необходимый для постоянного мониторинга и отслеживания деятельности учреждения по вопросам студенческих общежитий. Это позволяет быстро и эффективно обслуживать студентов, оптимизировать работу с документами, устранить злоупотребления, своевременно принимать ответственные решения и многое другое.

Целью моего дипломного проекта является – создание системы принятия и обработки заявлений на заселение в общежития, а также анализ и минимизация рисков для высших учебных заведений.

В ходе работы были использованы методы исследования, такие как: интервьюирование иногородних студентов и сотрудников университетов, изучение и обобщение отечественной практики, а также для разработки информационной системы был проведен анализ литературы.

Основным преимуществом системы цифровизации является возможность быстро получать практически всю информацию, необходимую для принятия решений, а также минимизировать ошибки в документации и многое другое.

1 Обзор и анализ предметной области

1.1 Эффективность внедрения информационных систем

По источникам из открытых данных, острый дефицит в решении жилищного вопроса на период обучения испытывают 17,4 тысяч иногородних студентов по всему Казахстану.

Основными причинами внедрения системы цифровизации являются: необходимость повышения качества и скорости обработки заявлений на заселение.

Основной проблемой, от которой страдают не только университеты, но и студенты, является скорость процесса подачи заявления на общежитие, а также обработка самих заявлений. Спасение от этого - внедрение эффективной системы цифровизации.

Во-первых, это связано с отсутствием возможности подачи заявления онлайн. На данный момент подавать заявления необходимо непосредственно в самом университете, а это влечет за собой необходимость приезжать в город, где находится университет, раньше начала учебного семестра и искать себе временное место проживания на период принятия, обработки заявления и публикации решения о заселении.

Во-вторых, есть вероятность в совершении ошибки сотрудником университета при обработке данных, так как работа монотонная и связана с документацией и большим количеством текстовой информации. А также в связи с огромным потоком заявлений есть вероятность утери документов.

Конечно, даже самая совершенная система цифровизации не решит все проблемы на 100%. Однако при правильном применении она может помочь минимизировать дефицит в жилищном вопросе на момент обучения среди студентов, а университету использовать студенческое общежитие только для основного его целевого действия. Сама по себе цифровизация – это эффективный инструмент управления учреждением и процессами, особенно контроля сотрудников.

При цифровизации процесса приема документов и данных при заявлении в общежитие сотрудник вряд ли перепутает цифры в индивидуальном идентификационном номере студента, который он получает из документов. Обработка заявлений через систему ускоряется.

1.2 Анализ существующих аналогов информационных систем

Университеты КазГУ, КазНАУ, АТУ и КазНМУ уже внедрили систему «Электронное общежитие» и принимают документы и заявления в электронном виде.

Интерфейс системы «Электронное общежитие» университета АТУ представлен на рисунке 1.

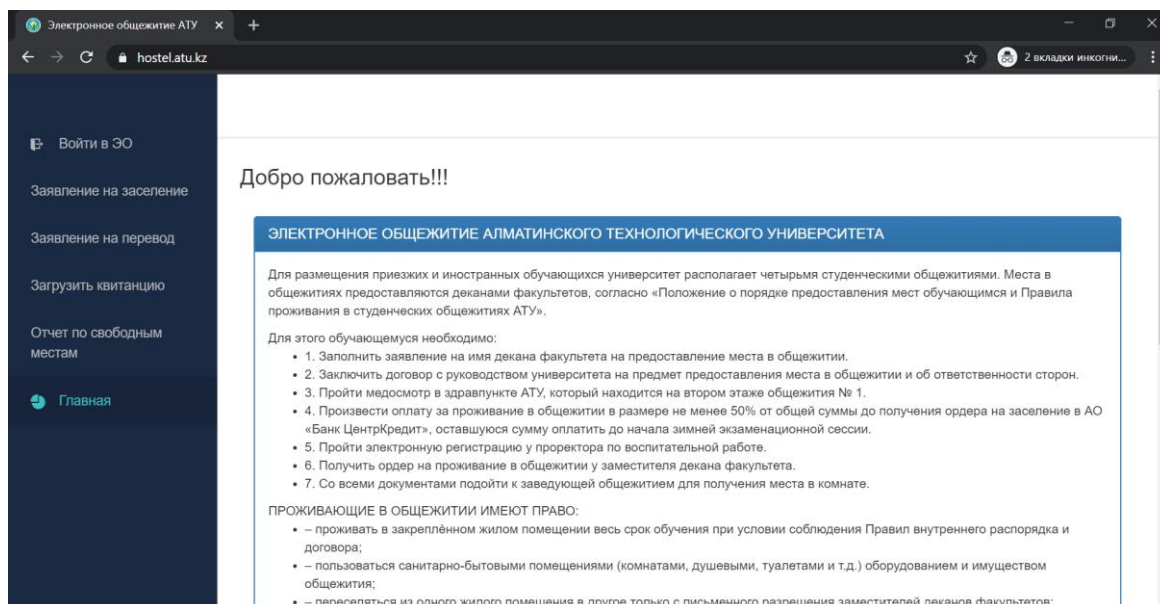


Рисунок 1.2.1 – интерфейс системы электронного общежития АТУ

Университет АТУ в 2013 году разработал систему единого электронного общежития, с целью внедрения его во все вузы Казахстана, чтобы вести единый список студентов, заселенных в общежитие. К данным системы имеют доступ только декан, сотрудники, отвечающие за студенческие общежития и родители студентов. В системе с легкостью можно найти информацию о заселенном студенте по ИИН: узнать адрес общежития, его комнату, соседей, ответственного за комнату, курс и так далее.

Сразу есть возможность прочитать и ознакомиться с порядком заселения, с правилами общежития, подать заявление из своего города, нет необходимости в физическом присутствии в комиссии по приему заявлений, есть возможность загрузить квитанцию об оплате.

1.3 Постановка задачи

Задача дипломного проекта заключается в создании системы принятия заявлений на заселение в общежитие. Система будет разрабатываться для НАО «Алматинский университет энергетики и связи», где на 2019-2020 учебный год насчитывается 5438 студента, из которых 4090 иногородние (75%). Университет имеет 3 общежития с общим количеством 1621 койко-место. Так как количество иногородних студентов существенно превышает количество мест в общежитиях имеет место цифровизации процесса заселения.

Сейчас в университете работает система приоритетности при заселении: сначала заселяются первокурсники, дальше заселяются студенты, имеющие льготы – это дети-инвалиды, сироты, дети, которые остались без попечения родителей и так далее, а уже потом рассматриваются заявления студентов со второго по четвертый курс.

Система будет реализовываться в формате сайта, к которой будет подключена база данных, на основе языков программирования HTML, CSS, JS, PHP.

Система будет состоять из страницы регистрации для внесения всей необходимой информации:

- полное ФИО;
- ИИН;
- номер телефона;
- email;
- пол;
- национальность;
- область;
- место прописки, адрес;
- факультет;
- специальность;
- курс;
- льготы;
- загрузка архива с необходимыми справками, документами;
- форма обучения;
- примечание.

Страница реализована как web-site для удобства внедрения ее на основную страничку университета aues.kz.

Чтобы реализовать данный проект необходимо изучить бизнес-процессы обеих сторон и осуществить следующие требования:

- анализ процесса заселения в университетское общежитие;
- выбор оптимальных инструментов для разработки;
- разработка интерфейса с учетом пользовательского опыта;
- разработка информационной системы.

2 Проектирование системы и обоснование выбора инструментов

На сегодняшний день процесс цифровизации общепита набирает обороты и ведется активная работа в данном направлении. В данном разделе описываются процессы проектирования и какие технологии были использованы при разработке дипломного проекта, их ключевые достоинства и недостатки, почему были выбраны именно эти технологии, и за какую часть они отвечают.

2.1 Проектирование системы принятия заявлений

Проектирование можно начать с планирования пути студента, которая включает описание типичных взаимодействий между пользователями системы и самой системой и предоставление описания ее работы.

На рисунке 2.1.1 представлена путь взаимодействия студента по отношению к университету.

Система наделяет их следующими функциональными возможностями:

- подача заявления;
- отправка документов.



Рисунок 2.1.1 – Алгоритм действий студента

На рисунке 2.1.2 представлены дальнейшие действия от университета, после принятия всех заявлений. Заявления принимаются только в определенные даты. Решения о заселении высылаются после нескольких дней после того, как закрывается прием документов. Система должна обеспечивать следующие функциональные возможности для сотрудника университета:

- выгрузку базы данных;
- просмотр данных;
- возможность сортировки списка заявлений по приоритетности.



Рисунок 2.1.2 – Алгоритм действий после сбора заявлений

Серверная часть сайта должна содержать базу данных, а также программные функции для реализации отправки данных.

На рисунке 2.1.3 можно представлены таблицы базы данных в количестве 18 штук, в них хранятся данные о пользователях, время подачи заявления, архивы с документами и т.д.

База данных была сделана при помощи phpMyAdmin. В целом для создания всей платформы дополнительно были приобретены домен и виртуальный хостинг. Благодаря виртуальному хостингу получаем удобную систему для управления базой данных и всей серверной части платформы.

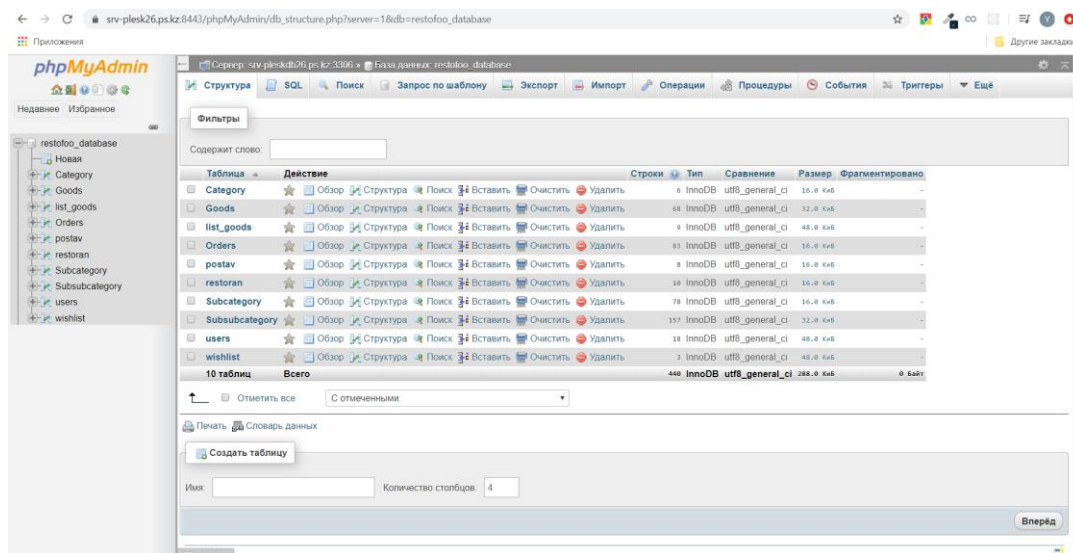


Рисунок 2.1.3 – Таблицы базы данных

Для развертывания сайта на домене был взят адрес aues-dorm.kz, который представлен на рисунке 2.1.4.

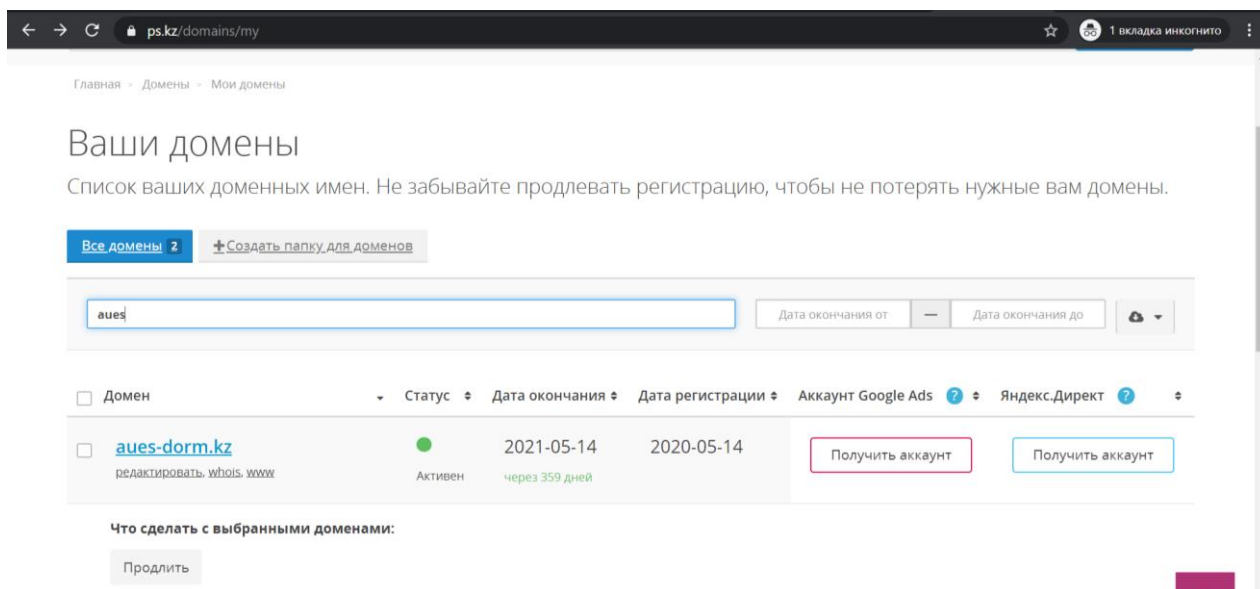


Рисунок 2.1.4 – Доменное имя нашей системы

Для демонстрации проекта был приобретен домен, который представлен на рисунке 2.1.4.

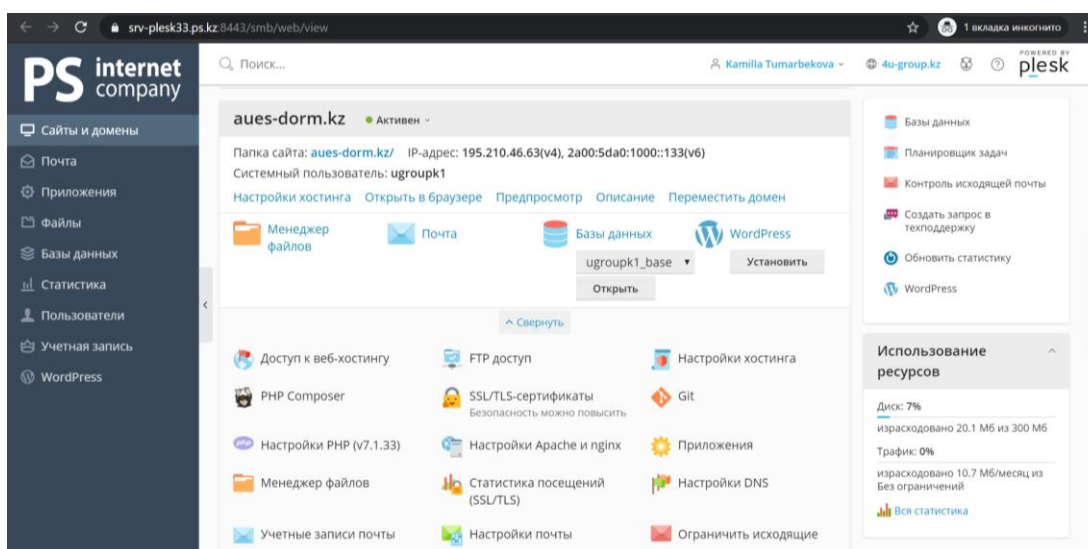


Рисунок 2.1.5 – Система управления серверной частью

На рисунке 2.1.5 представлена система управления серверной частью сайта, которая называется Plesk. Используя Plesk можно администрировать исходные файлы проекта, управлять базой данных, настраивать сетевые параметры, вести статистику.

На рисунке 2.1.6 представлен код, реализующий отправку данных, вводимых в форму, в базу данных.

```

add.php x  setting.php x
1 <?php
2
3 if ($_POST['enter'] and $_POST['name'] and $_POST['surname'] and $_POST['patronymic'] and $_POST['email'] and $_POST['phone'] and $_POST['comment'] and $_POST['iin'] and
4 $_POST['geo'] and $_POST['gender'] and $_POST['nazi'] and $_POST['course'] and $_POST['form_edu'] and $_POST['stage'] and $_POST['specialty'] and $_POST['privileges']) {
5 $_POST['name'] = FormChars($_POST['name']);
6 $_POST['surname'] = FormChars($_POST['surname']);
7 $_POST['patronymic'] = FormChars($_POST['patronymic']);
8 $_POST['email'] = FormChars($_POST['email']);
9 $_POST['phone'] = FormChars($_POST['phone']);
10 $_POST['comment'] = FormChars($_POST['comment']);
11 $_POST['iin'] = FormChars($_POST['iin']);
12 $_POST['geo'] = FormChars($_POST['geo']);
13 $_POST['nazi'] = FormChars($_POST['nazi']);
14 $_POST['gender'] += 0;
15 $_POST['course'] += 0;
16 $_POST['form_edu'] += 0;
17 $_POST['stage'] += 0;
18 $_POST['specialty'] += 0;
19 $_POST['privileges'] += 0;
20
21 if ($_FILES['file']['tmp_name']) {
22 if ($_FILES['file']['type'] != 'application/octet-stream') MessageSend(2, 'Не верный тип файла.');
```

Рисунок 2.1.6 – Отправка данных в базу

2.2 Выбор инструментов для разработки платформы

Для разработки предложенного проекта был использован стандарт последней версии-HTML5 (HyperText Markup Language). HTML5 - это обновление, сделанное в HTML из HTML4 (XHTML следует другой схеме нумерации версий). Он использует те же основные правила, что и HTML 4, но добавляет некоторые новые теги и атрибуты, которые позволяют улучшить семантику и динамические элементы, активируемые с помощью JavaScript.

Впервые разработанный Тимом Бернерсом-Ли в 1990 году, HTML — это сокращение от языка разметки гипертекста (HyperText Markup Language). HTML используется для создания электронных документов (называемых страницами), которые отображаются во Всемирной паутине. Каждая страница содержит ряд соединений с другими страницами, называемых гиперссылками. Каждая веб-страница, которую вы видите в Интернете, написана с использованием той или иной версии HTML-кода.

HTML-код обеспечивает правильное форматирование текста и изображений для вашего интернет-браузера. Без HTML браузер не будет знать, как отображать текст в виде элементов или загружать изображения, или другие элементы. HTML также предоставляет базовую структуру страницы, на которую накладываются Каскадные Таблицы Стилей (CSS - Cascading Style Sheets) для изменения ее внешнего вида. Можно представить себе HTML как кости (структуру) веб-страницы, а CSS-как ее оболочку (внешний вид).

CSS - это язык таблиц стилей, который позволяет изменять внешний вид языка разметки. Например, HTML используется для создания базового макета веб-страницы, например, этого абзаца текста. CSS используется для определения шрифта, размера шрифта, веса шрифта, его положения и других визуальных настроек.

Сокращенно от cascading style sheets, CSS-это язык, используемый для описания многоцветных стилей представления документов, написанных на

языке разметки. Его концепция была создана Håkon Wium Lie в 1994 году. В декабре 1996 года CSS был внесен в спецификацию W3C и сегодня позволяет веб-разработчикам изменять макет и внешний вид своих веб-страниц. Например, CSS может использоваться для изменения шрифта, используемого в определенном элементе HTML, а также его размера и цвета. Один CSS-файл может быть связан с несколькими страницами, что позволяет разработчику изменять внешний вид всех страниц одновременно.

Также был использован для адаптации дизайна страницы инструмент Bootstrap. Это открытый и бесплатный HTML, CSS и JS фреймворк, который используется веб-разработчиками для быстрой вёрстки адаптивных дизайнов сайтов и веб-приложений.

Фреймворк Bootstrap представляет собой набор CSS и JavaScript файлов. На рисунке 2.2.1 представлен вид библиотеки Bootstrap, подключенной к нашей системе.

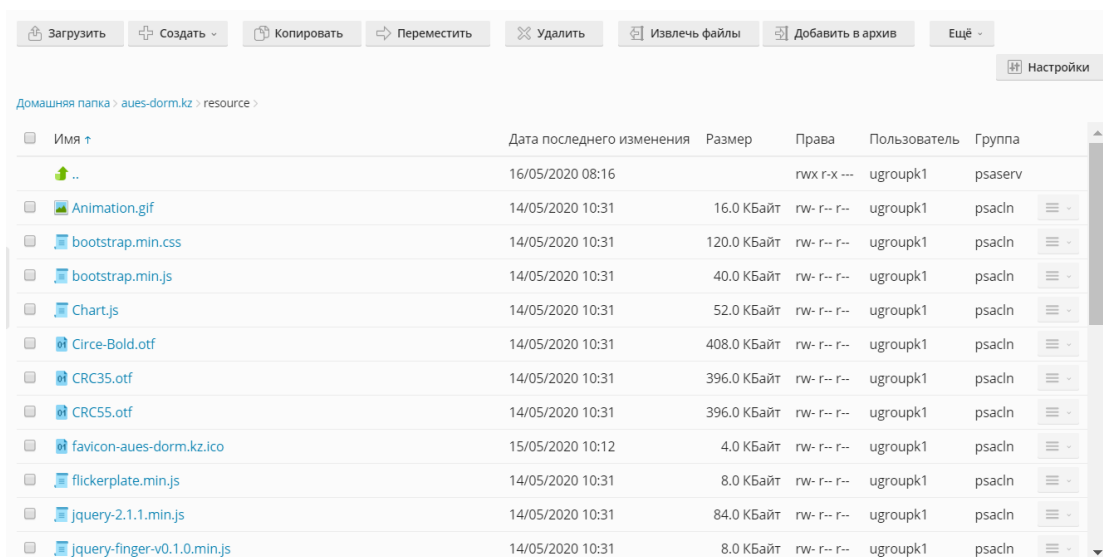


Рисунок 2.2.1 – Библиотека файлов и стилей Bootstrap

Bootstrap состоит из:

- системы сеток;
 - классов для стилизации текста, изображений, таблиц и другого контента;
 - компонентов предназначенных для создания кнопок, различных форм на странице, горизонтальных и вертикальных навигационных меню, слайдеров, выпадающих списков, аккордеонов, модальных окон, всплывающих подсказок и других элементов интерфейса;
 - классов для решения вспомогательных задач наиболее часто возникающими перед веб-разработчиками (выравнивание текста, скрытие или отображение элемента, задания цвета и фона элементу, задание margin и padding отступов, и др.).
- Bootstrap позволяет ускорить адаптацию вида страницы под основные электронные устройства пользователей.

По опыту в среднем 80% пользователей заходят на сайты через мобильные устройства.

Перед началом работы был изучен язык программирования PHP, шаблоны и методики программирования. Выяснилось, что PHP в общем-то без особых проблем позволяет реализовывать те или иные паттерны разработки. То есть в PHP можно писать правильный код, который будет не сильно отличаться от того, что получаем на Python/Java.

Проще говоря, PHP — это аббревиатура от слова «hypertext preprocessor». Это наиболее часто используемый серверный скриптовый язык, который используется преимущественно для веб-разработки. PHP является наиболее широко и широко используемым серверным сайтом в интернете, поскольку он, как известно, управляет или охватывает почти 82 % всего интернета сегодня. Помимо его использования в веб-разработке, PHP также используется в качестве языка программирования общего назначения. Компьютерный язык PHP был впервые использован в истории в 1995 году и, как известно, был разработан за год до этого великим компьютерным ученым Расмусом Лердофом. Первоначально PHP обозначал «личную домашнюю страницу», которая в последующие годы была заменена на «гипертекстовый препроцессор».

Помимо разработки веб-сайтов, PHP используется с целью разработки веб-программ. Одна очень уникальная и контрастная особенность компьютерного языка PHP заключается в том, что его скрипты могут быть интерпретированы только на серверах, на которых уже установлен PHP. Еще одна контрастная особенность языка PHP заключается в том, что компьютеры клиентов, которые фактически получают доступ к веб-сайту, разработанному с использованием PHP, не требуют установки PHP на своем компьютере, им просто требуется веб-браузер для доступа к нему. PHP совместим в основном с Windows, а также с Unix-подобными операционными системами.

В нашем проекте применен JavaScript для разработки модальных окон, таких как: восстановление пароля, оформление заказа, редактирование товара. Он был выбран, так как скрипты не требуют перезагрузки страницы, что упрощает для пользователей работу с системой. JavaScript - это, как известно клиентский скрипт. Большинство веб-приложений, таких как поисковая система, работают благодаря взаимодействию между устройством пользователя (например, компьютером, телефоном или планшетом) и удаленным сервером. Программное обеспечение на удаленном сервере отправляет информацию клиенту (т. е. машине пользователя), а программное обеспечение на стороне клиента считывает информацию и выводит веб-страницу на экран.

JavaScript - это язык программирования, используемый в основном веб-браузерами для создания динамического и интерактивного контента для пользователя. Большинство функций и приложений, которые делают Интернет незаменимым в современной жизни, закодированы в той или иной форме JavaScript.

Самые ранние воплощения JavaScript были разработаны в конце 1990-х годов для веб-браузера Netscape Navigator. В то время веб-страницы были статичными, предлагая мало взаимодействия с пользователем, кроме щелчка по ссылкам и загрузки новых страниц. Впервые JavaScript включил анимацию, адаптивное содержимое и проверку формы на странице.

В течение многих лет JavaScript функционировал только в ограниченном количестве браузеров. Explorer от Microsoft, крупнейший базовый браузер стал поддерживать JavaScript намного позже. Вместо этого Microsoft создала свой собственный клиентский скрипт под названием JScript. В первые дни веб-разработки программисты, которые хотели создавать динамические веб-сайты, часто были вынуждены выбирать одно семейство браузеров вместо другого. Это было далеко не идеально, поскольку делало Интернет менее общедоступным.

Для разработки базы данных была выбрана СУБД MySQL, так как при приобретении виртуального сервера, MySQL является системой управления реляционными базами данных по умолчанию и ввиду того, что наша система требует особых условий к скорости работы БД, масштабируемости и др., то есть можем использовать MySQL ввиду её высокой популярности, относительно высокой скорости работы, наличие большого количества документации и большого сообщества разработчиков.

MySQL-это поддерживаемая Oracle система управления реляционными базами данных с открытым исходным кодом, основанная на языке структурированных запросов (SQL). MySQL работает практически на всех платформах, включая Linux, UNIX и Windows. Хотя он может быть использован в широком спектре приложений. MySQL основан на модели клиент-сервер. Ядром MySQL является сервер MySQL, который обрабатывает все инструкции базы данных (или команды). Сервер MySQL доступен как отдельная программа для использования в сетевой среде клиент-сервер и как библиотека, которая может быть встроена (или связана) в отдельные приложения.

MySQL работает вместе с несколькими служебными программами, которые поддерживают Администрирование баз данных MySQL. Команды отправляются на сервер MySQL через клиента MySQL, который установлен на компьютере. MySQL изначально был разработан для быстрой обработки больших баз данных. Хотя MySQL обычно устанавливается только на одной машине, он способен отправлять базу данных в несколько местоположений, поскольку пользователи могут получить к ней доступ через различные клиентские интерфейсы MySQL. Эти интерфейсы отправляют инструкции SQL на сервер, а затем отображают результаты. MySQL позволяет хранить данные и получать к ним доступ через несколько механизмов хранения, включая InnoDB, CSV и NDB. MySQL также способен реплицировать данные и разбивать таблицы на разделы для повышения производительности и долговечности. Пользователи MySQL не обязаны изучать новые команды; они могут получить доступ к своим данным с помощью стандартных команд SQL.

Для работоспособности и использования платформы было приобретено доменное имя и виртуальный хостинг с панелью управления Plesk, благодаря которой можно управлять базой данных и всеми файлами платформы.

Parallels Plesk Panel или просто Plesk вкратце известен как одна из самых многофункциональных панелей управления на планете. Это выходит далеко за рамки возможностей обычных веб-инструментов для управления серверными ресурсами, необходимыми для существования ваших сайтов. Plesk - это полноценная многоуровневая панель управления хостинговой платформой, предоставляющая администратору сервера широкий спектр возможностей графического интерфейса для настройки и обслуживания всего хостингового бизнеса, обслуживающего конечного клиента. Он так же хорош для настройки общих хостинговых решений, как и для настройки виртуальных и выделенных серверов. Операционные системы, поддерживающие Plesk, включают Fedora, Red Hat, Debian, FreeBSD, SuSE, CentOS, Ubuntu и Windows.

Панель управления Plesk может похвастаться простым в навигации интерфейсом автоматизации веб-хостинга. Навигация по его многочисленным разделам была максимально облегчена, так что вы могли бы быстро найти необходимую вам функциональность в любой момент времени. Панель управления запоминает все наиболее часто используемые функциональные возможности и приносит их вам в любое время, когда вы получаете доступ к ее интерфейсу.

3 Разработка программного продукта

Серверная часть приложения не имеет собственного визуального представления и взаимодействует с пользователем через веб-браузер. Название этой части происходит от того, что все действия выполняются на сервере — специальном компьютере, который может находиться за тысячи километров от браузера или рядом с машиной. Сервер обычно размещает базу данных и использует такие языки, как Java, PHP, C# и т. д. Это приложение разработано на языке программирования PHP.

Прежде чем разрабатывать клиентскую часть нужно продумать процессы всех сторон и начать разработку.

3.1 Системные требования

Рекомендуемые системные требования:

- операционная система: Windows 7/8/10 — 64-bit.
- процессор (CPU): Intel Core i3.
- ОЗУ (RAM): 6 ГБ (или больше).
- свободное место на жёстком диске: ~35 МВ.

Минимальные системные требования:

- операционная система: Windows 7/8/10 — 64-bit.
- процессор (CPU): Intel Pentium.
- ОЗУ (RAM): 512 МВ (или больше).
- свободное место на жёстком диске: ~35 МВ.

3.2 Описание разработанных функций

Для написания программной части был применен язык разметки HTML, PHP, JavaScript, а также для придания стилей был применен CSS и библиотека Bootstrap. Для развертывания информационной системы был приобретен домен и виртуальный хостинг, а также применена система управления файлами сайтами и базой данных.

Вначале юзер проходит регистрацию через соответствующую форму. Далее после заполнения полей данные отправляются для обработки на сервер, где попадают в таблицу БД.

Первоначально пользователь заносит свои данные в форму подачи заявления на заселение по ссылке aues-dorm.kz. Вёрстка вида формы представлен фрагментом на рисунке 3.2.1.

```

main.php x style.css x index.php x
9
7 <div class="container-fluid block-add">
8 <div class="container block-add-2">
9 <div class="row">
10 <div class="col-md-1"></div>
11 <div class="col-md-10 block">
12 <div class="col-md-4 logo">
13 <p class="text-center"></p>
14 </div>
15 <div class="col-md-8 block-content">
16 <p class="p-left">Заявление на заселение</p>
17 <form method="POST" class="custom_input" action="/proposals/add" enctype="multipart/form-data">
18 <p>
19 <span class="podskazki">Полное ФИО</span> <br>
20 <input type="text" id="redex" class="form-control custom_input_input" name="surname" placeholder="Бава фамилия" required> <br>
21 <input type="text" id="redex" class="form-control custom_input_input" name="name" placeholder="Бава имя" required> <br>
22 <input type="text" id="redex" class="form-control custom_input_input" name="patronymic" placeholder="Бава отчество" required>
23 </p>
24 <p><span class="podskazki">ИИН</span> <br> <input id="redex" type="text" name="iin" class="custom_input_input form-control iin-mask"
placeholder="___-___-___" required> <div class="form-check">
25 </p>
26 <p><span class="podskazki">Ваш номер телефона</span> <br> <input id="redex" type="text" class="custom_input_input form-control phone-mask"
name="phone" placeholder="+7 (___) ___-___" required></p>
27 <p><span class="podskazki">Ваш email</span> <br> <input id="redex" class="custom_input_input form-control " type="email" name="email"
placeholder="info@mail.com" required></p>
28 </div class="row">

```

Рисунок 3.2.1 – Фрагмент кода вида странички

Важно учитывать, что данные номер телефона, ИИН имеет определенную форму заполнения. ИИН состоит из 12 цифр, а формат номера должен быть един, а именно +7 (___) ___-___ для удобства при дальнейших рассылках через сторонние сервисы. Для того, чтобы у пользователя не было возможности ошибиться в количестве цифр или формате отправки данных установим на эти части формы специальные маски. Код для реализации масок для минимизирования проблем с заполнением данных представлена на рисунке 3.2.2.

```

111
112 <script src="resource/jquery.maskedinput.min.js" type="text/javascript"></script>
113 <script src="resource/jquery.maskedinput.js" type="text/javascript"></script>
114 <script>
115     $(document).ready(function() {
116     $(".phone-mask").click(function(){
117         $(this).setCursorPosition(3);
118     }).mask("+7 (999) 999-99-99");
119     });
120
121 </script>
122 <script>
123     $(document).ready(function() {
124     $(".iin-mask").click(function){
125         $(this).setCursorPosition(3);
126     }).mask("999-999-999-999");
127     });
128
129 </script>
130

```

Рисунок 3.2.2 – Скрипт реализации масок на форме заявления

Также для минимизации проблем с ошибками, которые могут достучить пользователи при заполнении данных вручную, большую часть информации мы ограничили с помощью инструмента выпадающего меню. Например, в университете есть только три формы обучения: очное, заочной и дистанционное, в университете есть только три степени: бакалавр,

магистратура, докторантура и так далее. Реализация представлена на рисунке 3.2.3.

```
main.php × style.css × index.php × add.php ×
46 <div class="row">
47 <div class="col-md-4">
48 <span class="podskazki">Курс</span> <br> <select size="1" id="redex" class="custom_input_select form-control" name="course" required>
49 <option value="1">1</option>
50 <option value="2">2</option>
51 <option value="3">3</option>
52 <option value="4">4</option>
53 </select>
54 </div>
55 <div class="col-md-4">
56 <span class="podskazki">Форма обучения</span> <br> <select size="1" class="custom_input_select form-control" id="redex" name="form_edu"
required>
57 <option value="1">Очное</option>
58 <option value="2">Заочное</option>
59 <option value="3">Дистанционное</option>
60 </select>
61 </div>
62 <div class="col-md-4">
63 <span class="podskazki">Степень</span> <br> <select size="1" id="redex" class="custom_input_select form-control" name="stage" required>
64 <option value="1">Бакалавриат</option>
65 <option value="2">Магистрант</option>
66 <option value="3">Докторант</option>
67 </select>
68 </div>
69 </div>
```

Рисунок 3.2.3 – Код выпадающего меню

Для загрузки документов, справок и заявления реализована кнопка выбрать файл. Её формат ограничен архивом формата .rar. При загрузке архива в базу данных архив преобразовывается в формат .zip в целях уменьшения количества занимаемого места на сервере.

```
95 <p><span class="podskazki">Загрузите необходимые справки и документы</span> <br> <span style="float: left; font-size: 12px;">все документы
загружаются в папку и сюда загружается архив в формате .rar</span> <br> <input type="file" class="custom_input_input form-control" name="file"></p>
96
97 <p>
```

Рисунок 3.2.4 – Фрагмент кода кнопки загрузки файлов

```
21 if ($_FILES['file']['tmp_name']) {
22 if ($_FILES['file']['type'] != 'application/octet-stream') MessageSend(2, 'Не верный тип файла.');
```

```
23 $_POST['link'] = 0;
24 } else $num_file = 0;
25
26
27 $MaxId = mysqli_fetch_row(mysqli_query($CONNECT, 'SELECT max(`id`) FROM `proporsals`'));
28 if ($MaxId[0] == 0) mysqli_query($CONNECT, 'ALTER TABLE `proporsals` AUTO_INCREMENT = 1');
29 $MaxId[0] += 1;
30
31
32 if ($_FILES['file']['tmp_name']) {
33 foreach(glob('catalog/file/*', GLOB_ONLYDIR) as $num => $Dir) {
34 $num_file ++;
35 $Count = sizeof(glob($Dir.'/*.*'));
36 if ($Count < 250) {
37 move_uploaded_file($_FILES['file']['tmp_name'], $Dir.'/'.$MaxId[0].'.zip');
38 break;
39 }
40 }
41 }
```

Рисунок 3.2.5 – Фрагмент кода проверки формата файла и преобразование его в формат .zip

На рисунке 3.2.5 представлен код подключения базы данных к форме подачи заявления.

```
main.php × style.css × index.php × add.php × setting.php ×
1 <?php
2 define ('HOST', 'srv-pleskdb33.ps.kz');
3 define ('USER', 'ugroupk1_base');
4 define ('PASS', 'Aues2020');
5 define('DB', 'ugroupk1_base');
6
7 ?>
```

Рисунок 3.2.6 – Код подключения базы данных

Подключение реализовано в отдельном файле для удобства переноса базы на личный сервер учреждения для повышения сохранности личных данных студентов.

После отправки данных студентом, то есть нажатия кнопки «Подать заявку» ему будет выходить рор-уп окошко с подтверждением принятия заявления на заселение и с дальнейшими действиями. Код реализации окошка подтверждения представлен на рисунке 3.2.7.

```
main.php × style.css × index.php × add.php × setting.php ×
26
27
28 }
29
30
31 if (in_array($Page, array('main'))) include("page/$Page.php");
32
33
34 else if ($Page == 'proporsals' and in_array($Module, array('add'))) include("module/proporsals/$Module.php");
35
36 function MessageSend($p1, $p2, $p3 = '', $p4 = 1) {
37 $_SESSION['message'] = '<script type="text/javascript" src="/resource/popup.js"></script><div class="popup_notice popup_style_2">
38 <p class="p-modal-main">Благодарим за Ваше заявление</p>
39 <p class="p-modal-content">заявления принимаются с 17 по 21 августа </p>
40 <p class="p-modal-content">решение о заселении придет вам на указанную в заявлении почту 23 августа 2020 года</p>
41 <p class="p-modal-content">по вопросам обращайтесь по номеру<br>
42 +7 (727) 237 89 88
43 </p>
44
45
46 </div>
47
48 '.$p2.'
49
```

Рисунок 3.2.7 – Код окошка подтверждения

3.3 Описание клиентской части

HTML состоит из серии коротких кодов, набранных в текстовый файл автором сайта - это теги. Затем текст сохраняется в виде html-файла и просматривается через браузер, например Internet Explorer, Google Chrome, Mozilla Firefox. Этот браузер читает файл и переводит текст в видимую форму, надеюсь, что страница будет выглядеть так, как задумал автор. Написание собственного HTML - кода влечет за собой правильное использование тегов для создания вашего видения. Для создания HTML-

страниц можно использовать что угодно-от примитивного текстового редактора до мощного графического редактора. Создать документы, написанные на языке HTML, которые станут основой вашего веб-сайта, можно несколькими способами. Теги - это то, что отделяет обычный текст от HTML-кода. Вы можете знать их как слова между <угловыми скобками>. Они создают все интересные вещи, такие как изображения, таблицы и прочее, просто сообщая вашему браузеру, что нужно визуализировать на странице. Разные теги будут выполнять разные функции. Сами теги не появляются, когда вы просматриваете свою страницу через браузер, но их эффекты появляются.

Самым распространенным способом является текстовый набор HTML-кода в текстовом редакторе. Это довольно трудоемкая работа, которая подразумевает широкое познание принципов разметки документов в HTML, навыки работы с каскадными таблицами стилей CSS, а также как минимум с JavaScript, PHP и фреймворками.

Вторым способом является использование CMS. Система управления контентом, часто сокращенно называемая CMS - это программное обеспечение, которое помогает пользователям создавать, управлять и изменять контент на веб-сайте без необходимости специальных технических знаний. Проще говоря, система управления контентом - это инструмент, который помогает вам создавать веб-сайт без необходимости писать весь код с нуля (или даже знать, как кодировать вообще). Вместо того чтобы создавать свою собственную систему для создания веб-страниц, хранения изображений и других функций, система управления контентом обрабатывает все эти базовые инфраструктурные вещи для вас, чтобы вы могли сосредоточиться на более перспективных частях вашего веб-сайта. Помимо веб-сайтов, вы также можете найти системы управления контентом для других функций, таких как управление документами.

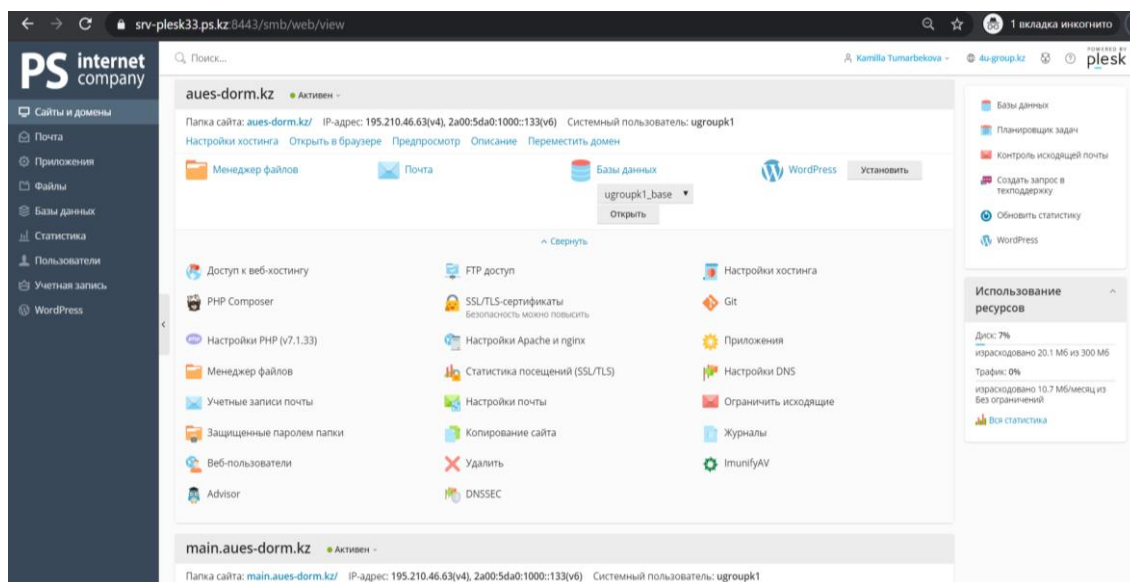


Рисунок 3.3.1 – Система управления

Как описано выше, также для управления клиентской частью сайта используется инструмент для менеджмента файлов, редктирования кода, а также множество других функций, которые входят в панель управления Plesk (рисунок 3.3.1).

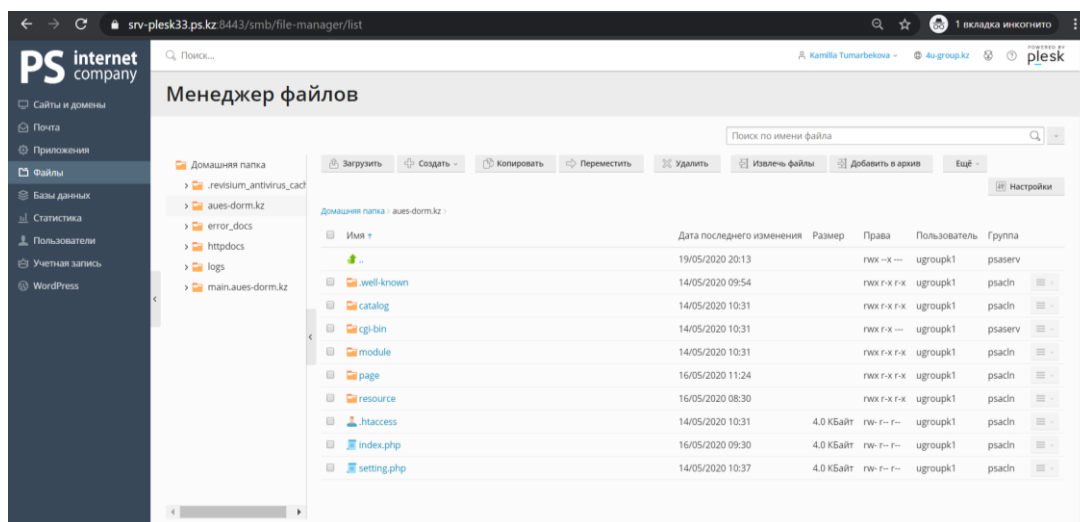


Рисунок 3.3.2 – Менеджер файлов сайта

На рисунке 3.3.2 представлен менеджер управления файлами системы, с помощью него можно комфортно работать со всеми файлами и редактировать их внутренний код.

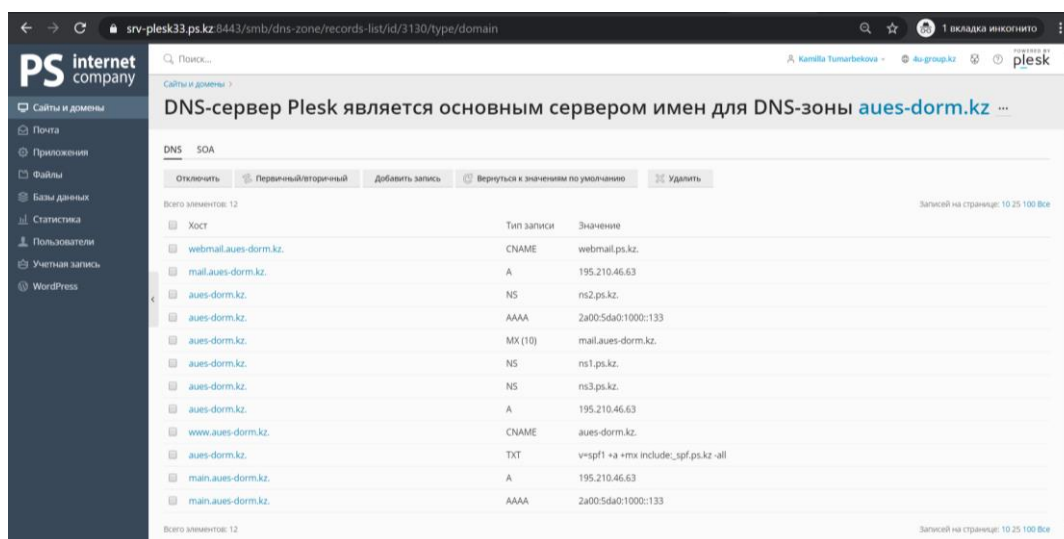


Рисунок 3.3.3 – Настройки DNS

Страница с функцией управления DNS записями представлена на рисунке 3.3.3. Здесь отображаются NS, MX, TXT записи.

Также через систему Plesk можно обезопасить свое доменное имя SSL- и TSL- сертификатами.

SSL является аббревиатурой для Secure Sockets Layer. Это тип цифровой безопасности, которая позволяет зашифровать связь между веб-сайтом и веб-браузером. Технология в настоящее время устарела и полностью заменена TLS.

Что такое TLS? Это означает Transport Layer Security и обеспечивает конфиденциальность данных так же, как и SSL. Поскольку SSL фактически больше не используется, это правильный термин, который люди должны начать использовать.

Что такое HTTPS? Это безопасное расширение HTTP. Веб-сайты, устанавливающие и настраивающие SSL/TLS-сертификат, могут использовать протокол HTTPS для установления безопасного соединения с сервером.

Цель SSL/TLS - сделать соединение безопасным для передачи конфиденциальной информации, включая личные данные, информацию о платеже или регистрации.

Это альтернатива простой передаче текстовых данных, в которой ваше соединение с сервером не зашифровано, и это затрудняет мошенникам и хакерам отслеживание соединения и кражу ваших данных.

Большинство людей знают, что такое SSL/TLS. Это сертификаты, которые используются веб-мастерами для защиты своих веб-сайтов и обеспечения безопасного доступа для людей к транзакциям.

Вы можете определить, использует ли веб-сайт сертификат безопасности, потому что рядом с URL-адресом в адресной строке появится значок маленького замочка.

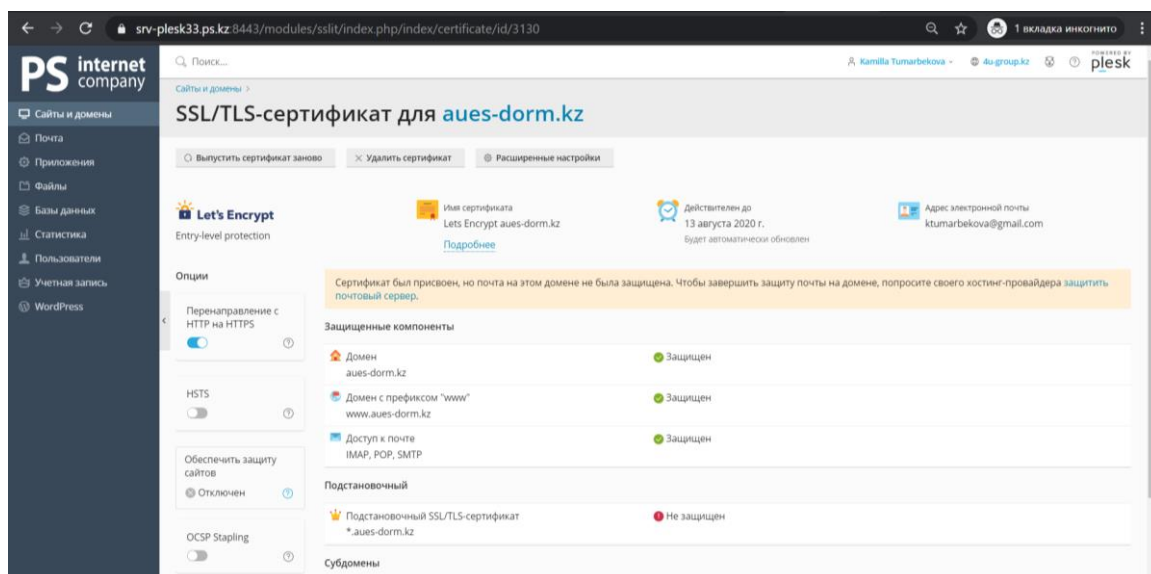


Рисунок 3.3.4 – Подключение SSL/TSL-сертификатов

На рисунке 3.3.4 представлена страничка, на которой можно подключить предлагаемые сертификаты. SSL/TLS является обязательным,

когда передаётся конфиденциальная информация, такая как имена пользователей и пароли или информация о платёжной обработке.

Цель SSL/TLS состоит в том, чтобы убедиться, что только один человек - лицо или организация, утверждённое пользователем, может получить доступ к передаваемым данным. Это особенно важно, когда вы думаете о том, между сколькими устройствами и серверами передаются данные до того, как они достигнут своего пункта назначения.

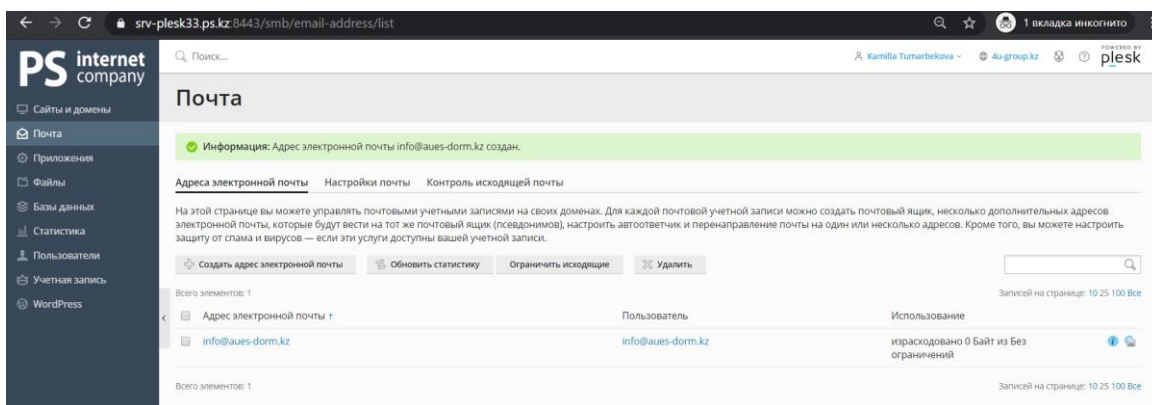


Рисунок 3.3.5 – Учетные записи почты

На странице, представленной на рисунке 3.3.5 вы можете управлять почтовыми учетными записями на своих доменах. Для каждой почтовой учетной записи можно создать почтовый ящик, несколько дополнительных адресов электронной почты, которые будут вести на тот же почтовый ящик (псевдонимов), настроить автоответчик и перенаправление почты на один или несколько адресов.

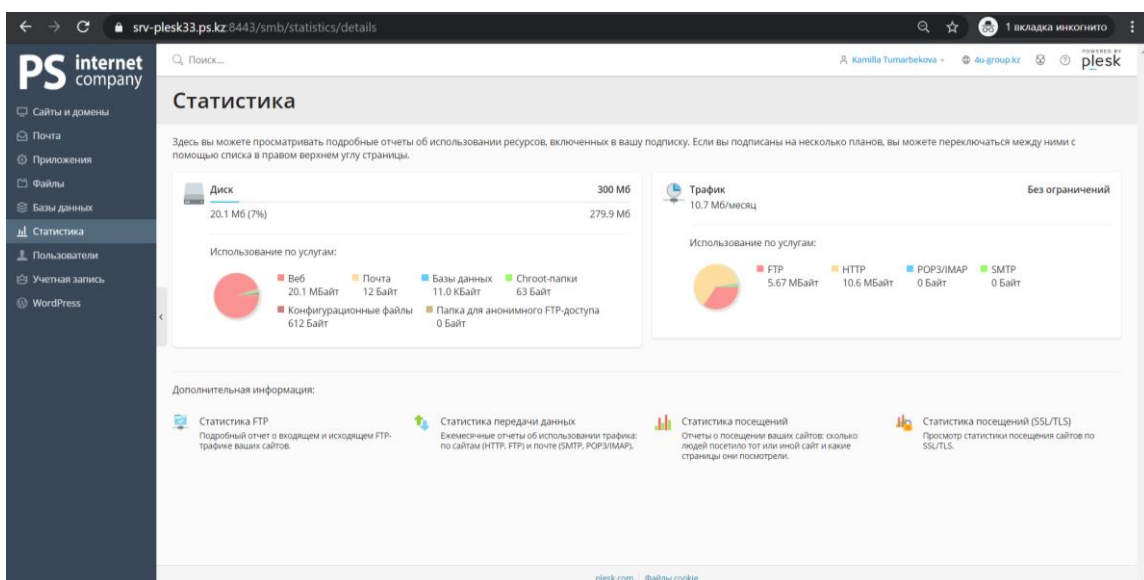


Рисунок 3.3.6 – Статистика использования сайта

На рисунке 3.3.6 представлен просмотр подробных отчетов об использовании ресурсов, например сколько место занято на виртуальном диске.

Также есть возможность создавать веб-сайты с помощью web-конструкторов. Создание веб-сайта раньше требовало обучения кодированию, но те времена остались в прошлом. Теперь любой желающий может легко создать новый сайт с помощью конструктора веб-сайтов. Независимо от того, владеете ли вы небольшим бизнесом или хотите создать свой собственный блог, создатели сайтов делают это так, что вам не нужно нанимать веб-дизайнера, чтобы иметь профессионально разработанный сайт.

Но слышать, что конструкторы сайтов просты в использовании, не означает, что вы просто автоматически знаете, что такое конструктор сайтов или как он работает. Создатели веб-сайтов позволяют пользователям легко создавать свой собственный веб-сайт без особых усилий или навыков кодирования. Используя эту платформу, любой желающий может создать профессионально разработанный сайт без использования веб-дизайнера.

Для написания внешнего вида элементов применяется каскадные таблицы стилей CSS.

CSS расшифровывается как Каскадные Таблицы Стилей с акцентом на «стиль». В то время как HTML используется для структурирования веб-документа (определяя такие вещи, как заголовки и абзацы, а также позволяя вам вставлять изображения, видео и другие медиа), CSS определяет стиль вашего документа-макеты страниц, цвета и шрифты определяются с помощью CSS. CSS привносит стиль на ваши веб-страницы, взаимодействуя с элементами HTML. Элементы - это отдельные HTML-компоненты веб-страницы - например абзац.

CSS имеет довольно простой синтаксис. Правила объявлений очень просты, и можно описать их в одном предложении. Во-первых пишется селектор, который выбирает конкретный элемент на странице после фигурных скобок пишутся свойства со значениями после двоеточия, а сами свойства отделяются друг от друга точкой с запятой. Самое сложное в CSS объявлении – селектор.

CSS селектор – (от слова select – выбирать) – это конструкция, с которой начинается каждый блок, и которая служит для выборки элемента или одинаковых элементов на странице для стилизации в будущем. Селекторы CSS используются для выбора контента, который вы хотите стилизовать. Селекторы являются частью набора правил CSS. Селекторы CSS выбирают HTML-элементы в соответствии с их идентификатором, классом, типом, атрибутом и т. д.

Важно учитывать, что, когда пользователь заходит на сайт, его взгляд в первую очередь падает на дизайн. Дизайн играет очень важную роль, потому что от этого будет зависеть захочет ли юзер просматривать страницу дальше

или закрывает её. Для хорошего проектирования, разработки сайта и определения аудитории часто UX/UI.

UX относится к любому взаимодействию пользователя с продуктом или услугой. UX design учитывает каждый элемент, который формирует, как он заставляет пользователя чувствовать себя, и насколько легко для пользователя выполнить свои желаемые задачи. Это может быть что угодно - от того, как физический продукт ощущается в вашей руке, до того, насколько прост процесс оформления заказа при покупке чего-то онлайн. Цель UX design - создать легкий, эффективный, актуальный и всесторонний приятный опыт для пользователя.

Дизайн пользовательского интерфейса (UI) - это процесс создания интерфейсов в программном обеспечении или компьютеризированных устройствах с акцентом на внешний вид или стиль. Дизайнеры стремятся создавать проекты, которые пользователи найдут простыми в использовании и приятными. Дизайн пользовательского интерфейса обычно относится к графическим пользовательским интерфейсам, но также включает в себя и другие, например, управляемые голосом.

На нашей платформе интерфейс должен быть простым и понятным, так как данным сайтом будут пользоваться люди из областных центров города в возрасте от 17 до 50 лет, включая студентов и их родителей, у которых, как все мы знаем, разный уровень пользования электронными гаджетами

Сайт будет состоять из главной страницы и модальных окон.

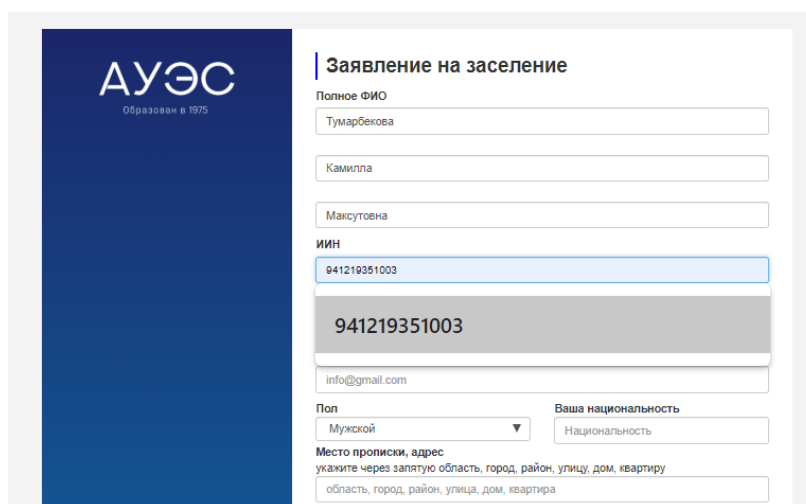
Был выбран синий цвет, который является корпоративным, а также по психологии цвета – синий цвет ассоциируется с знаниями. Интуитивно поля для ввода данных просты и понятны, а также дополнительно имеют подсказки.

The image shows a web browser window with the URL 'aues-dorm.kz'. The page title is 'Заявление на заселение в общежитии'. The form is titled 'Заявление на заселение' and contains the following fields and sections:

- Имя Ф.И.О.**: Fields for 'Ваше фамилия', 'Ваше имя', and 'Ваше отчество'.
- ИИН**: Field for 'Ваш ИИН'.
- Ваш номер телефона**: Field for 'Ваш номер телефона' with a '+7' prefix.
- Ваш email**: Field for 'Ваш email'.
- Пол**: Dropdown menu with 'Мужской' selected.
- Ваша национальность**: Dropdown menu with 'Казак' selected.
- Место проживания, адрес**: Text field with a placeholder 'укажите через запятую область, город, район, улицу, дом, квартиру'.
- Курс**: Dropdown menu with '1' selected.
- Форма обучения**: Dropdown menu with 'Онлайн' selected.
- Специальность**: Dropdown menu with '88011000 - Информатика' selected.
- Льготы**: Dropdown menu with 'Льготы отсутствуют' selected.
- Загрузите необходимые сканы и документы**: Text field with a placeholder 'или сканните QR-код и загрузите документы в формате jpg'.
- Выберите файл**: Button for file selection.
- Дополнительный комментарий/примечание**: Text area for additional comments.
- ПОДАТЬ ЗАЯВКУ**: Submit button.

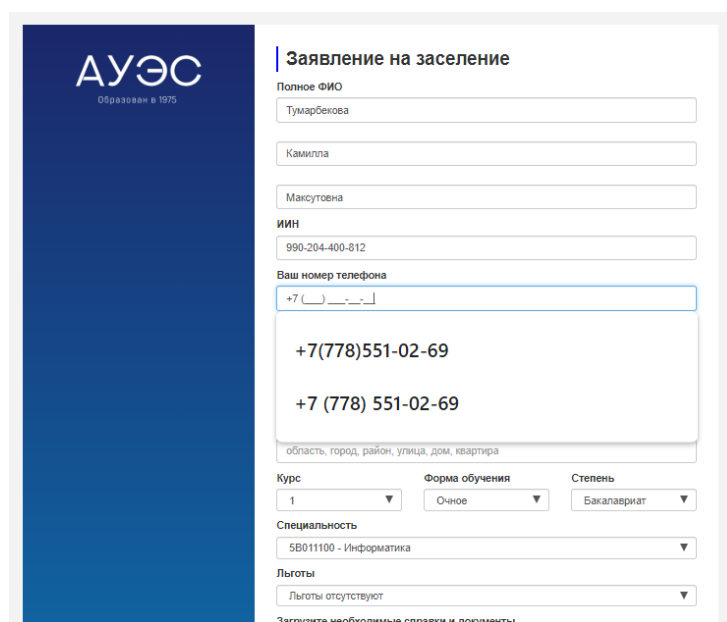
Рисунок 3.3.7 – Вид страницы формы заявления на заселение

На рисунках 3.3.8 и 3.3.9 представлен визуальный вид наложенной маски на поля ввода «ИИН» и «Номер телефона». Сделано это из-за необходимости точности этих данных, ведь номер телефона люди могут заполнять через «8» или через «+7», но нам нужно, чтобы все номера телефоном были именно через «+7», для удобства производить информационные рассылки.



The screenshot shows a web form titled "Заявление на заселение" (Application for settlement) for the organization "АУЭС" (AUES), established in 1975. The form includes several input fields: "Полное ФИО" (Full name) with pre-filled values "Тумарбекова", "Камилла", and "Максумовна"; "ИИН" (IIN) with a mask "941219351003" and a dropdown menu showing the same value; "Пол" (Gender) with a dropdown menu set to "Мужской" (Male); "Ваша национальность" (Your nationality) with a dropdown menu set to "Национальность" (Nationality); and "Место прописки, адрес" (Place of registration, address) with a dropdown menu. The IIN field is highlighted with a blue border, indicating the mask is active.

Рисунок 3.3.8 – Вид работы маски на поле ИИН



The screenshot shows the same web form as in Figure 3.3.8, but with a mask applied to the "Ваш номер телефона" (Your phone number) field. The mask is "+7 () - - - -". The dropdown menu shows two options: "+7(778)551-02-69" and "+7 (778) 551-02-69". The form also includes fields for "Курс" (Course) set to "1", "Форма обучения" (Form of study) set to "Очное" (Full-time), "Степень" (Degree) set to "Бакалавриат" (Bachelor's), "Специальность" (Specialty) set to "5B011100 - Информатика" (5B011100 - Informatics), and "Льготы" (Benefits) set to "Льготы отсутствуют" (Benefits are absent). A note at the bottom says "Загрузите необходимые справки и документы" (Upload necessary certificates and documents).

Рисунок 3.3.9 – Вид работы маски на поле номера телефона

Отправить форму, не заполнив хотя бы одного поле, невозможно. Так как в коде стоит скрипт на проверку пустых обязательных полей. При попытке отправить пустую форму заявителю выходит напоминание «Заполните это поле», как продемонстрировано на рисунке 3.3.10.

Пол: Мужской

Ваша национальность: казашка

Место прописки, адрес: укажите через запятую область, город, район, улицу, дом, квартиру
 область, город, район, улица, дом, квартира

Курс: 1

Степень: Бакалавриат

Специальность: 5В011100 - Информатика

Льготы: Льготы отсутствуют

Сообщение об ошибке: Заполните это поле.

Рисунок 3.3.10 – Вид проверки на пустые поля

После успешной подачи заявления, заполнения всех нужных данных из формы в качестве подтверждения получения данных выходит рор-уп окошко.

Pop up окно – это всплывающее окно, открываемое на экране компьютера в результате выполнения какой-либо операции. Для создания использовалась технология JavaScript. Оно создано для того, чтобы минимизировать количество дублей при создании таблицы данных.

АУЭС
Образован в 1975

Заявление на заселение

Полное ФИО

Благодарим за Ваше заявление

заявления принимаются с 17 по 21 августа

решение о заселении придет вам на указанную в заявлении почту 23 августа 2020 года

по вопросам обращайтесь по номеру
+7 (727) 237 89 88

Ваш email: info@gmail.com

Пол: Мужской

Ваша национальность: Национальность

Рисунок 3.3.11 – Вид рор-уп окошка подтверждения принятия заявления

4 Технико-экономическое обоснование

Темой дипломного проекта является «Разработка системы «Электронное общежитие» для высшего учебного заведения».

В данной дипломной работе описан проект по разработке системы сбора заявлений на заселение в студенческое общежитие, разработана конфигурация для высшего учебного заведения.

Целью данного проекта является – обеспечение университета системой сбора информации об иногороднем студенте, который подает заявление на заселение, и распределением мест с учетом привилегий студента. Данная конфигурация облегчает труд сотрудников университета, перекладывая многие задачи на компьютер, а также имеет в себе ведение базы данных.

В разработке программного продукта примет участие качественно подобранная команда IT-специалистов, состоящая из таких работников, как: технический оператор, develop-программист. Технический оператор осуществляет и следит за правильностью заполнением данных, контролем нужных документов и в первую очередь за качество будет отвечать именно технический оператор. Develop-программист в свою очередь занят разработкой технической части проекта, разработкой программной части приложения, всеми сопутствующим этому тестами и готовит систему с самого начального и до конечного выпуска продукта в свет. Соответственно, весь основной процесс разработки осуществляет программист, на плечи которого и ложится большая часть задач, пока как технический оператор совершает дела вне программного осуществления, а именно сопутствующая этому организационная работа.

Технико-экономическое обоснование имеет следующие пункты:

- определение сложности разработки программного обеспечения;
- расчет затрат на разработку ПО;
- определение ценности готового продукта;
- оценка результатов работы программного обеспечения.

4.1 Определение сложности разработки ПО

Для того, чтобы точно определить сложность разработки программного обеспечения, необходимо произвести деление всей задачи на более простые этапы. Это позволит эффективно следить за прогрессом разработки программного обеспечения, за счет деления сложной задачи на более легкие подзадачи. Такой подход, с моей точки зрения, считается более эффективным и позволяет результативно и быстро обрабатывать подзадачи. Модель распределения сложности разработки ПО и стадии разработки представлены в таблице 4.1.

Таблица 4.1 - Этапы разработки ПО

Этапы разработки ПО	Вид работы	Трудоемкость, чел. час.
Этап 1	Постановка задач	20
Этап 2	Разработка и утверждение ТЗ на разработку ПО	30
Этап 3	Поиск и изучение подобных программ	15
Этап 4	Поиск и изучение сопутствующей литературы	20
Этап 5	Составление аналитических графиков ПО	15
Этап 6	Оформление теоретической части дипломной работы	25
Этап 7	Разработка практической части дипломного проекта	20
Этап 8	Реализация проекта	50
Этап 9	Отладка и устранение недоработок	20
Этап 10	Организация отчета и результатов работы	15
Этап 11	Тестирование ПО	30
Этап 12	Подведение итогов по разработанному ПО	10
Этап 13	Внедрение	25
Итого: трудоемкость выполнения дипломного проекта		295

Продолжительность рабочего дня равна 8 часам. В результате для реализации программного обеспечения необходимо 37 рабочих дней.

4.2 Расчет затрат на разработку ПО

Определение затрат необходимых для разработки программного обеспечения производится на основе имеющейся сметы, которая включает следующие элементы:

- материальные затраты;
- затраты на оплату труда;
- социальный налог;
- амортизация основных фондов;
- прочие затраты.

Материальные затраты делятся на основные и вспомогательные затраты на материалы, энергию и другие затраты необходимые для разработки ПО. Расчет материальных затрат происходит по форме, предоставленной в таблице 4.2.

Таблица 4.2 – Затраты на материальные ресурсы

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Бумага для офиса	Svetocopy	Пачка	3	1200	3600
Тетрадь (96 листов)	ErichCruse	Штук	2	200	400
Блокнот	ErichCruse	Штук	2	450	900
Ручки	Maxriter	Штук	2	200	400
Компьютерная мышь	Logitech	Штук	1	3000	3000
Итого:					8 300

Для разработки программного обеспечения будет использоваться ноутбук Asus VivoBook S330UN, мощности ноутбука достаточно для выполнения поставленных задач. Ввиду того, что ноутбук содержит установленную операционную систему и программное обеспечение необходимое для разработки ПО, нет нужды производить дополнительные расходы на новые ОС и ПО.

Общую сумму, необходимую на материальные средства (Z_M) можно рассчитать по следующей формуле:

$$Z_M = \sum P_i * C_i, \quad (4.1)$$

где P_i - расход i -го вида материального ресурса, натуральные единицы;
 C_i - цена за единицу i -го вида материального ресурса, тг;
 i - вид материального ресурса;
 n - количество видов материальных ресурсов.

Расчет затрат на необходимое оборудование и программное обеспечение производится по форме, приведенной в таблице 4.3.

Таблица 4.3 – Расчет затрат на оборудование и ПО, необходимое для проекта

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Ноутбук	Asus VivoBook S330UN	Штук	1	396000	396000
Принтер	Samsung	Штук	1	58000	58000

	SCX-3400				
<i>Продолжение таблицы 4.3</i>					
Модем	TP-LINK TD	Штук	1	17 000	17 000
ОС	Windows 10	Штук	1	-	-
Итого:					471000

$$З_{\text{м}} = 8\,300 + 471\,000 = 479\,300 \text{ (тг)}$$

Для реализации программного обеспечения необходимы материалы на сумму 479 300 тенге.

4.3 Расчет затрат на электроэнергию

Так как при разработке программного обеспечения не обойтись без потребления электроэнергии, имеет смысл произвести расчет затрат на электроэнергию.

Согласно таблице 4.1 для разработки программного обеспечения необходимо порядка 295 часов, теперь необходимо рассчитать стоимость электроэнергии, которая будет потрачена в течение всего времени работы. Для принтера расчет будет проводиться для периода в 24 часа, так как нет необходимости постоянно использовать принтер.

$$\mathcal{E} = \mathcal{E}_{\text{эл.эн.обор.}} + \mathcal{E}_{\text{доп.нужды.}} \quad (4.2)$$

где $\mathcal{E}_{\text{эл.эн.обор.}}$ – затраты на электроэнергию оборудования;

$\mathcal{E}_{\text{доп.нужды.}}$ – затраты электроэнергии на дополнительные нужды.

Расчет электроэнергии, которая необходима для оборудования определяется по следующей формуле:

$$\mathcal{E}_{\text{эл.эн.обор.}} = \sum W * K_{\text{исц}} * S * T, \quad (4.3)$$

где W – потребляемая мощность, Вт;

$K_{\text{исц}}$ – коэффициент использования ($K_{\text{исц}} = 0,7..0,9$);

T – время работы;

S – тариф (1кВт/ч = 19,17 тенге).

Итоги по расчетам стоимости затрачиваемой электроэнергии представлены в таблице 4.4.

Таблица 4.4 - Затраты на электроэнергию

Наименование приборов	Паспортная мощность, кВт	Коэффициент мощности	Время работы оборудования, ч	Цена ЭЭ тг/кВтч	Сумма, тг.
-----------------------	--------------------------	----------------------	------------------------------	-----------------	------------

Ноутбук	0,6	0,8	350	19,17	3220,5
---------	-----	-----	-----	-------	--------

Продолжение таблицы 4.4

Модем	0,08	0,9	350	19,17	483,08
Принтер	0,5	0,9	24	19,17	207
Кондиционер	0,8	0,9	180	19,17	2484,4
Освещение	0,3	0,8	350	19,17	1610,2
Итого:	8004				

$$Z_{\text{эл.эн.обор.}} = 8004 \text{ (тенге)}$$

На дополнительные потребности расходы подсчитываются на основе повышенного показателя в объеме 5% от расходов на электроэнергию:

$$Z_{\text{доп.нужды}} = 5\% * Z_{\text{эл.эн.обор.}} \quad (4.4)$$

Определим затраты на дополнительные потребности согласно формуле (4.4):

$$Z_{\text{доп.нужды}} = 0.05 * 8004 = 400,2 \text{ (тенге)}$$

Исходя из всех расчетов, полные расходы на электроэнергию составляют:

$$Z = 400,2 + 8004 = 8404,2 \text{ (тенге)}$$

4.4 Расчет затрат на оплату труда

Для разработки программного обеспечения, как указывалось ранее, необходимо два работника:

- руководитель проекта – управление рабочим временем, корректировка рабочих процессов, координация, изучение предметной области;

- разработчик – разработка ПО, тестирование и сопровождение.

Сумму расходов на оплату труда можно рассчитать по следующей формуле:

$$Z_{\text{тр}} = \sum ЧС_i * T_i \quad (4.5)$$

где $ЧС_i$ - часовая ставка i -го работника, тг;

T_i - трудоемкость разработки модели, чел.×ч; i - категория работника;

n - количество работников, занятых разработкой ПП.

Во время реализации проекта рабочее время участников не равномерно, поэтому имеет смысл установить часовую ставку каждого работника и общий объем заработной платы.

Часовую ставку сотрудника можно рассчитать по следующей формуле:

$$ЧС_i = \frac{ЗП_i}{ФРВ_i} \quad (4.6)$$

где $ЗП_i$ - месячная заработная плата i -го работника, тг;
 $ФРВ_i$ - месячный фонд рабочего времени i -го работника, час.

Месячная заработная плата руководителя равняется 190 000 тенге, и месячная заработная плата разработчика равняется 160 000 тенге. Рассчитаем часовую ставку каждого работника согласно формуле (4.6):

$$ЧС_{\text{руководитель}} = \frac{190\,000}{22 * 8} = 1\,079,54 \text{ тг/ч}$$

$$ЧС_{\text{разработчик}} = \frac{160\,000}{22 * 8} = 909,09 \text{ тг/ч}$$

Часовая ставка руководителя составляет 1 079,54 (тг/ч), трудоемкость разработки равняется 100 часам. Часовая ставка разработчика составляет 909,09 (тг/ч), трудоемкость разработки равняется 295 часам. Согласно формуле (4.5) можно рассчитать сумму расходов на заработную плату работников:

$$З_{\text{тр}} = 1079,54 * 100 + 909,09 * 295 = 107954 + 268181 = 376\,135$$

Расчеты затрат по оплате труда показаны в таблице (4.5).

Таблица 4.5. – Расчет заработной платы

Категория работника	Квалификация	Трудоемкость разработки ПП, час.	Часовая ставка, тг/ч	Сумма, тг.
Руководитель	Проектный руководитель	100	1079,54	107954
Разработчик	Программист	225	909,09	318181
Итого:				426 135

4.5 Расчет затрат по социальному налогу

Согласно Налоговому кодексу Республики Казахстан, социальный налог составляет 9,5% от фонда оплаты труда. Налоги, уплаченные за работников юридическим лицом, можно рассчитать по следующей формуле:

Таблица 4.6 – Формулы для расчета затрат по социальному налогу

Уплаченные налоги юридическим лицом	10,46	ФОТ	426 135
-------------------------------------	-------	-----	---------

Продолжение таблицы 4.6

СО (Социальные отчисления)	3,5	(ЗП - ОПВ)*3,5%	13 423,25
ВОСМСЮ (Отчисления на ВОСМСЮ)	2,0	ЗП*2%	8 522,70
СН (Социальный налог)	9,5	(ЗП - ОПВ - ВОСМСФ)*9,5%-СО	22 606,46
Всего уплаченные налоги			44 552,41

Результаты расчетов представлены в таблице (4,7):

Таблица 4.7 – Начисление социальных отчислений и налогов

Категория работника	Количество о человек	Заработная плата, тг	Социальные отчисления и налоги, тг
Руководитель	1	107954	11 286,59
Разработчик	1	318181	33 265,82
Итого:			44 552,41

4.6 Амортизация основных фондов и прочие затраты

Нормы амортизации ОФ необходимо определить в соответствии с налоговым кодексом РК. Амортизацию ОФ можно определить по следующей формуле:

$$A_r = \frac{C_{об} * H_a}{100} \quad (4.8)$$

где, $C_{об}$ – стоимость оборудования;

H_a – норма амортизации (норма амортизация = 25);

Формула (4.8) позволяет рассчитать нужную сумму для амортизационных отчислений за год для ноутбука:

$$A_r = \frac{300\,000 * 25}{100} = 75\,000 \text{ тенге}$$

Теперь необходимо рассчитать норму амортизации за период разработки:

$$A_r = \frac{62500 * 34}{365} = 6986,3 \text{ тенге}$$

Подобным образом необходимо рассчитать норму амортизации для всего оборудования. Результаты расчетов приведены в таблице (4.7).

Таблица 4.8 – Амортизация ОФ

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Сумма амортизации за год, тг	Сумма амортизации за время разработки, тг
Ноутбук	300 000	25	75 000	6986,3
Принтер	58 000	25	14 500	1350,57
Модем	17 000	20	2 800	316,6
Итого:			92 300	8652,07

Научные командировки не предусмотрены.

Расходы по статье «Прочие затраты» (P_{zi}) на конкретное ПО включают затраты на приобретение и подготовку специальной научно-технической информации и специальной литературы. Определяются по нормативу, разрабатываемому в целом по организации, в процентах к основной заработной плате:

$$P_{zi} = Z_{oi} \cdot \frac{H_{пз}}{100}$$

где $H_{пз}$ - норматив прочих затрат в целом по организации в (%), в дипломной работе нужно брать 20%.

$$P_z = 318181 \cdot 0,2 = 63\ 636,2 \text{ тенге}$$

Затраты по статье «Накладные расходы» (P_{ni}), связанные с необходимостью содержания аппарата управления, вспомогательных хозяйств и опытных (экспериментальных) производств, а также с расходами на общехозяйственные нужды (P_{ni}), относятся на конкретное ПО по нормативу ($H_{нр}$) в процентном отношении к основной заработной плате исполнителей. Норматив устанавливается в целом по организации:

$$P_{ni} = Z_{oi} \cdot \frac{H_{нр}}{100}$$

где P_{ni} - накладные расходы на конкретную ПО (тыс. тенге);

$H_{нр}$ - норматив накладных расходов в целом по организации в (%), в дипломной работе нужно брать 70%.

$$P_{ni} = 318181 \cdot 0,7 = 222\ 726,7 \text{ тенге}$$

Смета расходов на разработку ПО.

На основе всех представленных расчетов необходимо оформить смету расходов на разработку ПО согласно форме, которая приведена в таблице (4.9). На рисунке продемонстрирована диаграмма рабочих расходов.

Таблица 4.9 – Смета затрат на разработку ПО

Статьи затрат	Сумма, тг
Затраты на оборудование	471000

Продолжение таблицы 4.9

Затраты на материальные ресурсы	8300
Затраты на программное обеспечение	0
Затраты на оплату труда	426135
Социальные налоги	44552,41
Затраты на электроэнергию	8004
Амортизация основных фондов	6986,3
Прочие расходы	63636,2
Накладные расходы	222726,7
Итого по смете:	1251340,61



Рисунок 4.6.1 – Диаграмма затрат

4.7 Определение возможной (договорной) цены ПО

Стоимость программного обеспечения определяется на основе качества разработанного продукта, сроков его разработки и производительности продукта. Стоимость $Ц_d$ для программного обеспечения можно рассчитать по следующей формуле:

$$Ц_d = Z_{\text{нир}} \left(1 + \frac{P}{100} \right), \quad (4.9)$$

где $Z_{\text{нир}}$ – затраты на разработку программного обеспечения, тг;

P – средний уровень рентабельности ПО, (%). Данный параметр принят равным 40 %.

$$Ц_{д} = 1251340,61 \left(1 + \frac{40}{100} \right) = 1\,751\,876,9 \text{ тенге}$$

Далее необходимо определить стоимость реализации с учетом НДС, ставка НДС устанавливается законодательством РК. На 2019 года ставка НДС составляет 12%. Стоимость реализации учитывая НДС можно рассчитать по следующей формуле:

$$Ц_{р} = Ц_{д} + Ц_{д} * \text{НДС}, \quad (4.10)$$

$$Ц_{р} = 1\,751\,876,9 + 1\,751\,876,9 * 0,12 = 1\,962\,102,1 \text{ тенге}$$

Разработчик участвует в освоении ПО и несет соответствующие затраты, на которые составляется смета, оплачиваемая заказчиком по договору. Затраты на освоение определяются по нормативу ($H_0=10\%$) от себестоимости ПО в расчете на 3 месяца и рассчитываются по формуле:

$$P_{oi} = C_{ni} \cdot \frac{H_0}{100} = 1251340,61 \cdot 0,1 = 125\,134 \text{ тенге}$$

Затраты на сопровождение ПО (P_{ci}). Организация-разработчик осуществляет сопровождение ПО и несет соответствующие расходы, которые оплачиваются заказчиком в соответствии с договором и сметой на сопровождение. Затраты на сопровождение определяются по установленному нормативу ($H_c=20\%$) от себестоимости ПО (в расчете на год) и рассчитываются по формуле:

$$P_{ci} = C_{ni} \cdot \frac{H_c}{100} = 1251340,61 \cdot 0,2 = 250\,268 \text{ тенге}$$

Капиталовложения программного обеспечения с учетом затрат на освоение и сопровождение будет:

$$K = 1\,962\,102,1 + 125\,134 + 250\,268 = 2\,337\,504,311 \text{ тенге.}$$

Данную цену можно округлить до 2 337 505 тенге.

4.8 Сравнительный анализ эксплуатационных затрат до и после внедрения

Рассмотрим список расходов после установки программного обеспечения:

- основная заработная плата проектного руководителя и developer-программиста проекта включая отчисляемыми налогами;
- материальные ресурсы;
- амортизация основных фондов;
- накладные расходы.

Список расходов до внедрения:

- основная заработная плата заведующего складом, бухгалтера и 2 работников отдела закупа;
- материальные ресурсы;
- накладные расходы.

4.8.1 Расчет затрат до внедрения ПО

Таблица заработной платы и отчисляемых налогов представлена в таблице 4.8.1.1

Таблица 4.8.1.1 - Затраты на заработную плату до внедрения ПО

Сотрудник	Зарботная плата в месяц, тг	Зарботная плата в год, тг	Социальные отчисления и налоги, тг	Сумма
Директор общежития	200000	2 400 000	250920	2 650 920
Заведующий общежития	160000	1 920 000	200736	2 120 736
Администратор	100000 (200 000)	1 200 000 (2 400 000)	250920 (125040)	2 650 920 (48000)
Всего	460000	6720000	702 576	7 422 576

Расчет затрат на необходимое для продукта ПО и оборудование представлен в таблице 4.8.1.2

Таблица 4.8.1.2 – Оборудование и амортизационные отчисления, необходимое для работы общежития

Наименование и марка материала	Количество	Цена за ед.	Сумма в тенге	Годовые амортизационные отчисления
Ноутбук Lenovo IdeaPad S145	3	200000	600000	20000
Принтер Samsung SL-M2020W A4-W	1	54875	54875	18292
Компьютерная мышь беспроводная Logitech M90	3	3210	9630	3210
Итого				41 502

Общие накладные расходы вычисляются по формуле:

$$З_{Н.Р.} = 7\,422\,576 * 0,2 = 1\,484\,515 \text{ тенге.}$$

В итоге, общие эксплуатационные расходы до внедрения программного обеспечения будут равны,

$$З = 7\,422\,576 + 41\,502 + 1\,484\,515 = 8\,948\,593 \text{ тенге.}$$

4.8.2 Расчет затрат после внедрения ПО

Таблица заработной платы и отчисляемых налогов представлена в таблице 4.8.2

Таблица 4.8.2 - Затраты на заработную плату после внедрения ПО

Сотрудник	Заработная плата в месяц, тг	Заработная плата в год, тг	Социальные отчисления и налоги, тг	Сумма
Директор общежития	200000	2 400 000	250920	2 650 920
Заведующий общежития	160000	1 920 000	200736	2 120 736
Администратор	100000	1 200 000	125460	1 325 460
Всего	460000	5 520 000	577 116	6 097 116

Таблица 4.8.1.2 – Оборудование и амортизационные отчисления, необходимое для работы общежития

Наименование и марка материала	Количество	Цена за ед.	Сумма в тенге	Годовые амортизационные отчисления.
Ноутбук Lenovo IdeaPad S145	3	200000	600000	20000
Принтер Samsung SL-M2020W A4-W	1	54875	54875	18292
Компьютерная мышь беспроводная Logitech M90	3	3210	9630	3210
Итого				41 502

Общие накладные расходы рассчитываются по формуле:

$$З_{Н.Р.} = 6\,097\,116 * 0,1 = 609711 \text{ тенге.}$$

Итого, общие эксплуатационные расходы до внедрения программного обеспечения будут равны,

$$З = 6\,097\,116 + 41\,502 + 609\,711 = 7\,358\,041 \text{ тенге.}$$

Для более ясности предоставим все данные в сравнительную таблицу 4.8.2.4

Таблица 4.8.2.4 - Годовые эксплуатационные затраты

Статьи	До внедрения ПП	После внедрения ПП
Годовая заработная плата	7 422 576	6 097 116
Амортизационные отчисления	41 502	41 502
Накладные расходы	1484515	609711
Всего	8948593	6748329

Годовая экономия денежных средств будет равна согласно расчету:

$$8948593 - 6748329 = 2200264 \text{ тенге.}$$

Также определим суммарные выгоды от внедрения программного продукта.

Технико-экономические показатели позволяют рационально определить целесообразность разработки ПО и ее установки, а также взвесить реальную выгоду компании, как для разработчика системы, так и для ее пользователя.

Также мы можем определить срок окупаемости программного обеспечения в месяцах по формуле:

$$P_{\text{окуп}} = 12 * \frac{Z_{\text{разработка}}}{Z_{\text{годовая}}} \quad (4.21)$$

где $P_{\text{окуп}}$ – период окупаемости в месяцах;

$Z_{\text{разр.}}$ – затраты на разработку программы;

$Z_{\text{годовая}}$ – годовая экономия.

$$P_{\text{окуп}} = 12 \cdot \frac{2\,337\,505}{2\,200\,264} = 12 \cdot 1,06 = 12,8 \text{ мес.}$$

Предвидя годовую выгоду продукта и годовые эксплуатационные затраты с внедрением программного обеспечения можно определить коэффициент эффективности ($K_{\text{эф.}}$) по формуле:

Расчетный коэффициент экономической эффективности капитальных вложений составляет:

$$E_p = \frac{Z_{\text{уг}}}{K}$$

$$E_p = \frac{Z_{\text{уг}}}{K} = \frac{3\,200\,264}{2\,337\,505} = 0,94$$

Коэффициент эффективности указывает на соотношение сэкономленных тенге и потраченных. В заключении можно с уверенностью заявить о том, что на 1 затраченный тенге приходится 0,94 тенге прибыли.

Показатели сравнительной экономической эффективности от внедрения программного продукта

Наименование показателей	Значение
Условная годовая экономия затрат, тенге	2200264
Коэффициент экономической эффективности капитальных вложений (Ер)	0,94
Срок окупаемости капитальных вложений (Тр)	12,8

Внедрение данного программного продукта в рабочий процесс университета приносит экономию средств и цифровизацию рабочего процесса. Установка данного программного комплекса позволяет сократить штат сотрудников, что в следствии сокращает затраты на выплату налогов, зарплат и прочих денежных расходов. Соответственно, уменьшится объем технического оснащения, амортизационные отчисления и затраты на электроэнергию. Программный комплекс выгодно цифровизирует процесс, облегчив труд персонала и увеличив его КПД, к тому же уменьшив расходы предприятия.

Ожидаемый годовой экономический эффект составил 2 200 264 тенге. Приложение окупится в первые 12,8 месяцев использования.

5 Безопасность жизнедеятельности

В данной дипломной работе описывается разработка системы для приема заявлений на заселение в общежитие в режиме онлайн. Система «электронное общежитие» разрабатывается специально для НАО «Алматинский университет энергетики и связи» в целях упрощения процесса приема заявления на заселение и распределения койко-мест без человеческого вмешательства по приоритетности. Чтобы обеспечить безопасность сотрудников, которые разрабатывали данную систему и которые в дальнейшем будут её поддерживать, был проведен анализ потенциально опасных и вредных факторов в офисе, воздействующих на персонал, а также выполнены расчеты обеспечивающие комфортные условия труда.

5.1. Анализ потенциально опасных и вредных факторов в офисе, воздействующих на персонал.

В процессе трудовой деятельности на сотрудника офиса могут воздействовать опасные (вызывающие травмы) и природные (вызывающие заболевания) производственные факторы. Например, недостаточная освещенность рабочей зоны, электромагнитные излучения от компьютера, повышенный уровень шума, пожаробезопасность, электробезопасность и так далее.

Если описывать стандартное рабочее место одного сотрудника-программиста, который будет работать с разработанной системой, то это наличие персонального стационарного компьютера, принтер, стол, офисное кресло, тумбочка.

На рабочем месте должны быть предусмотрены меры по защите от возможного воздействия опасных и вредных производственных факторов. Уровни этих факторов не должны превышать предельных значений, предусмотренных правовыми, техническими и санитарно-техническими нормами. Эти нормативные документы обязывают создавать условия труда либо на рабочем месте, при которых влияние опасных и вредных факторов на работников устраняется полностью, либо в допустимых пределах.

К вредным и опасным физическим факторам, относящихся к труду офисных помещениях относятся:

- 1) Имеющие возможности передвижения офисного оборудования (тумбочки, кареты с документами).
- 2) Повышенная запыленность, загазованность воздуха рабочей зоны.
- 3) Повышенное значение напряжения в электрической цепи, замыкание которой может пройти через тело человека.
- 4) Повышенная напряженность магнитного поля.
- 5) Отсутствие и недостаток естественного света.
- 6) Недостаточная освещенность рабочей зоны.

При работе за компьютером сотрудник может ощущать дискомфорт: возникают головные боли и резь в глазах, появляются усталость и раздражительность. У некоторых людей нарушается сон, аппетит, ухудшается зрение, начинают болеть руки, шея, поясница и тому подобное. При ненормированной работе возможно нервное истощение.

В целях снижения пагубного влияния, опасные факторы, воздействию которых подвергается человек, работающий за ЭВМ, подлежат нормированию.

Основными нормируемыми параметрами являются:

- визуальные параметры мониторов;
- освещенность рабочего места;
- значения параметров электромагнитных излучений;
- оптимальные значения параметров микроклимата;
- уровень ионизации воздуха;
- уровень звука и звукового давления в октавных полосах частот;
- нормы вибрации

5.2. Расчет искусственного освещения

5.2.1. Исходные данные

Таблица 1 – Исходные Данные

Помещение	Офис
Ширина помещения (В), м.	6,5
Длина помещения (L), м.	12
Высота помещения, м.	3
Высота окна ($h_{\text{окна}}$), м.	2
Высота от пола до начала окна ($h_{\text{нач.окна}}$), м.	0,8
Высота рабочей поверхности ($h_{\text{раб.п.}}$), м.	0,8
Длина окна, м.	11
Расстояние до соседнего здания (Р), м.	40
Высота соседнего здания ($H_{\text{здан.}}$), м.	5
Расположение окон	Одностороннее, выходят на север

Таблица 2 – Исходные данные

Кол-во светильников	Расстояние от светильника до стены по длине помещения	Расстояние от светильника до стены по ширине помещения	Расстояние между светильниками по длине помещения	Расстояние между светильниками по ширине помещения
8 шт.	0,5 м.	1,5 м.	3 м.	3 м

Таблица 3 – Характеристика источника света

Напряжение питающей сети	Тип светильника	Кол-во ламп в светильнике	Лампы	Тип ламп
220В	ПВЛМ 2×40	2 шт.	Газоразрядные низкого давления	ЛД -40

Расчет искусственного освещения по методу коэффициента использования светового потока.

Необходимый световой поток, обеспечивающий создание требуемого нормированного общего освещения в помещении, определяется по формуле:

$$\Phi = \frac{E_{min} \cdot S \cdot Z \cdot K_3}{N \cdot \eta} \quad (2.1)$$

где: $K_3 = 1,5$ согласно (табл. 10) [1];

$E_{min} = 500$ лк III, а разряд зрительных работ (табл. 13) [1];

$S = 12 \times 6,5 = 78 \text{ м}^2$;

$Z = 1,25$ коэффициент минимальной освещенности (отношение средней и минимальной освещенности);

$N = 8$ шт.

Значение коэффициента использования зависит от индекса помещения i и коэффициентов отражения $\rho_{ст}$ стен, $\rho_{пот}$ потолка, $\rho_{пол}$ пола помещения, а так же высоты подвеса светильников h_p . Коэффициенты отражения стен, потолка и пола согласно табл. 15 [1] соответственно равны 73:60:30.

Высота подвеса светильников определяется как расстояние между уровнем рабочей горизонтальной поверхности и светильником:

$$h_p = H - (h_{раб.п.} + h_{св}) \quad (2.2)$$

где $h_{св}$ – расстояние между светильником и потолком (0 м)

$$h_p = 3 - (0,8 + 0) = 2,2 \text{ м.}$$

Индекс помещения определяется по формуле:

$$i = \frac{A \cdot B}{h_{расч} \cdot (A + B)} \quad (2.3)$$

$$i = \frac{12 \cdot 6,5}{2,2 \cdot (12 + 6,5)} = 1,9$$

Согласно таблице 5-11 [2] для светильника ПВЛМ 2×40 находим $\eta = 0,63$

Таким образом, световой поток равен:

$$\Phi = \frac{500 \cdot 78 \cdot 1,25 \cdot 1,5}{8 \cdot 2 \cdot 0,63} = 725 \text{ лм.}$$

Вывод: для обеспечения нормированного освещения в офисе необходимо расположить 8 светильников с минимальной мощностью светового потока каждого 725 люмен.

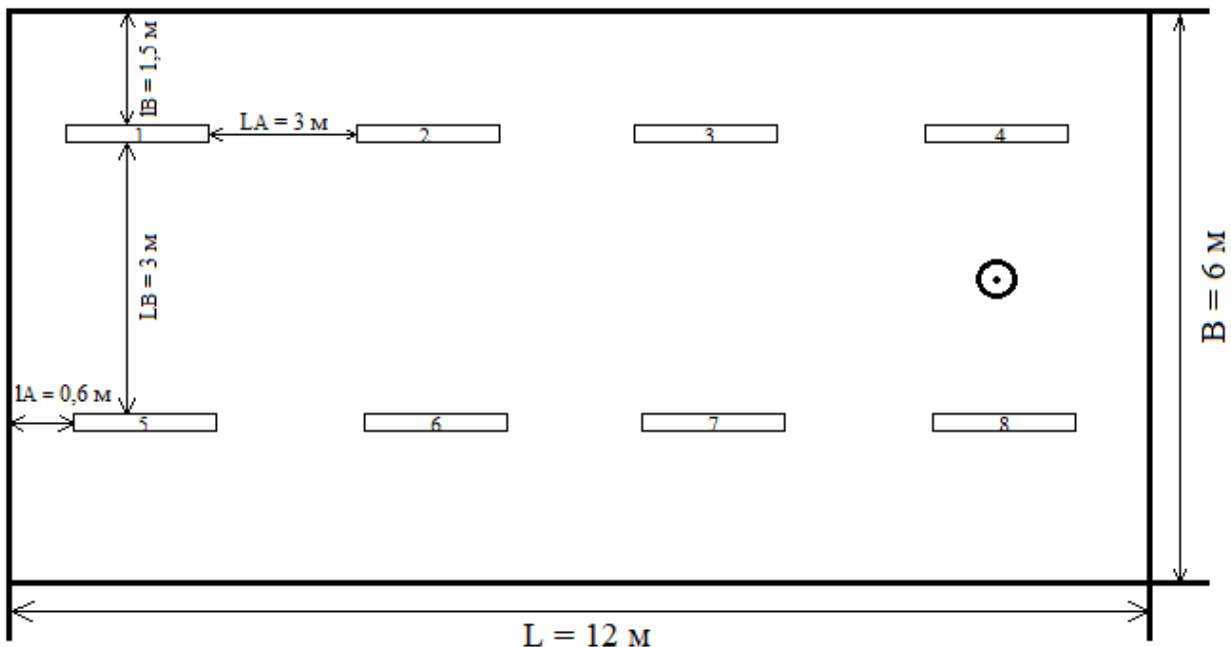


Рисунок 1 – Схема размещения светильников.

5.3 Расчет систем кондиционирования рабочего помещения IT отдела.

Расчет систем кондиционирования требуется для приведения в соответствие температурного режима в помещениях нормам и создать благоприятные условия для сотрудников.

Выводимый воздух из пространства $t_{\text{ВЫТ}}$, °С,

$$t_{\text{ВЫТ}} = t_{\text{РЗ}} + \Delta t \cdot (h - z) \quad (5.11)$$

где $t_{\text{РЗ}}$ – оптимальная температура для работы ($t_{\text{РЗ}} \leq t_{\text{ДОП}}$), °С;

Для теплого периода года $t_{\text{РЗ}} = 23$ °С;

h - расстояние от пола до вытяжки кондиционера на потолке $h = 3$ м;

Δt - температурный градиент по высоте помещения, °С;

z - высота рабочего помещения, $z = 2,7$ м.

$$t_{\text{ВЫТ}} = 23 + 1,4 \times (3 - 2,7) = 23,4 \text{ °С}$$

Итак, видим, что присутствует избыток явной теплоты. Поэтому для регулировки температурного режима и поддержке на благоприятном уровне, требуется подавать воздух ($t_{\text{ПР}}$) на 5-7°С ниже температуры воздуха в пространстве.

$$t_{\text{ПР}} = 23 - 7 = 16 \text{ °С}$$

Произвели калькуляцию количества приточного воздуха $L_{\text{ПР}}$, м³/ч:

$$L_{\text{ПР}} = \frac{Q_{\text{ИЗБ}}}{c \cdot \rho_{\text{ПР}} \cdot (t_{\text{ВЫТ}} - t_{\text{ПР}})} \quad (5.12)$$

где $\rho_{\text{ПР}}$ - 1,2 кг/м³ - плотность поступающего в помещение воздуха;
 c - 1кДж/(кг × °С) – удельная теплоемкость воздуха при постоянном давлении;

$t_{\text{ВЫТ}}$ - температура воздушной массы, удаляемой из рабочего пространства, °С;

$Q_{\text{ИЗБ}}$ – избыточное выделение явной теплоты, кДж/ч;

$t_{\text{ПР}}$ - температура приточного воздуха в рабочее помещение, °С.

Установили величину излишек выделения явной теплоты $Q_{\text{ИЗБ}}$:

$$Q_{\text{ИЗБ}} = \sum Q - \sum Q_{\text{УХ}} \quad (5.13)$$

где $\sum Q$ - суммарное количество поступающей явной теплоты из вне и выделяемой явной теплоты внутри помещения;

$\sum Q_{\text{УХ}}$ - суммарное количество выводящейся из пространства теплоты.

Тепловыделения от искусственного освещения Q_2 :

$$Q_2 = 1000 \times N \quad (5.14)$$

N – Мощность освещения, Вт.

$$Q_2 = 1000 \times 0,25 \times 20 = 5000 \text{ Вт}$$

Тепловыделение от людей Q_3 :

$$Q_3 = n \times q \quad (5.15)$$

где n - число сотрудников;

q - количество тепла, от каждого человека, Вт.

Таблица 5.2 - Количество тепла, от одного человека в зависимости от категории работ и температуры окружающей среды.

Категория выполняемых работ	Выделяемое тепло			
	Полное		Явное	
	10 °С	35 °С	10 °С	35 °С
Легкая	160 Вт	140 Вт	155 Вт	135 Вт

$$Q_3 = 21 \times 140 = 2940 \text{ Вт}$$

Количество тепла от солнечной радиации $Q_{\text{ОСТ.РАД}}$:

$$Q_{\text{ОСТ.РАД}} = F_{\text{ОСТ}} \times q_{\text{ОСТ}} \times A_{\text{ОСТ}} \quad (5.16)$$

Количество тепла от солнечной радиации в зависимости от наружного покрытия:

$$Q_{\text{П.РАД}} = F_n \times q_n \times k_n \quad (5.17)$$

где $F_{\text{ОСТ}}$ и F_n - площадь через которое поступает солнечная радиация, м^2 ;

$q_{\text{ОСТ}}$ и q_n - теплопоступления через 1 м^2 поверхности. коэффициент теплопередачи равен $1 \text{ Вт}/(\text{м}^2 \times \text{°С})$;

$A_{\text{ОСТ}}$ - коэффициент остекления рабочего помещения;

k_n - коэффициент теплопередачи покрытия, $\text{Вт}/(\text{м}^2 \times \text{°С})$.

$q_{\text{ОСТ}}$ - принимается в пределах 70 – 210;

$A_{\text{ОСТ}}$ – принимается в пределах 0,25 - 1,25.

$$F_{\text{ОСТ}} = 50 \text{ м}^2$$

$$q_{\text{ОСТ}} = 150 \text{ Вт}/(\text{м}^2 \times ^\circ\text{C});$$

$A_{\text{ОСТ}} = 0,4$ – окна в пространстве не затемненные;

$$Q_{\text{ОСТ.РАД}} = 50 \times 150 \times 0,4 = 3\,000 \text{ Вт}$$

$Q_{\text{П.РАД}} = 18 \text{ Вт}$ - среднее теплоступление покрытия;

$$Q_{\text{П.РАД}} = F_n \times q_n \times k_n$$

$$Q_{\text{П.РАД}} = 50 \times 1 \times 18 = 900 \text{ Вт}$$

Количество тепла, которое выходит из рабочего пространства $Q_{\text{УХ}}$, кВт:

$$Q_{\text{УХ}} = \frac{\lambda \cdot S \cdot (t_{\text{ВЫТ}} - t_{\text{ПР}})}{\delta} \quad (5.18)$$

где λ - теплопроводность стен, Вт/(м \times $^\circ\text{C}$);

S - площадь, м^2 ;

δ - толщина стен, м;

$\delta = 0,3\text{м}$ толщина стен, которые выполнены из бетона К750 с теплопроводностью 1,2 Вт/(м \times $^\circ\text{C}$).

$$Q_{\text{УХ}} = \frac{1,2 \cdot 330 \cdot (23,4 - 16)}{0,3} = 9768 \text{ Вт}$$

Сумма входящей теплоты в помещение:

$$\Sigma Q = Q_2 + Q_3 + Q_{\text{ОСТ.РАД}} + Q_{\text{П.РАД}} \quad (5.19)$$

$$\Sigma Q = 5000 + 2940 + 3000 + 90 = 11030 \text{ Вт}$$

Избыточное выделение явной теплоты:

$$Q_{\text{ИЗБ}} = 11\,030 - 9\,768 = 1\,262$$

Количество приточного воздуха необходимого для поддержания комфортных условий:

$$L_{\text{ПР}} = \frac{1262}{1 \cdot 1,2 \cdot (23,4 - 16)} = 142,12 \frac{\text{м}^3}{\text{ч}}$$

Так как мы видим присутствие избыточного выделения теплоты, произвели калькуляцию, что приведения нынешних условий труда в

благоприятные и соответствию нормативам необходимо произвести монтаж кондиционера.

Мы выбрали кассетный кондиционер LG CT24 - инверторный кассетный кондиционер со сплит – системой и возможностью как обогрева, так и охлаждения.

Таблица 5.3 Характеристики сплит-системы Gree-42

1	Мощность в режиме охлаждения	13 000 Вт
2	Мощность в режиме обогрева	14 600 Вт
3	Потребляемая мощность при охлаждении	3 200 Вт
4	Потребляемая мощность при обогреве	3 000 Вт
5	Уровень шума (мин/макс)	33/37 Дб
6	Обслуживаемая площадь	130 кв.м
7	Внутренний блок сплит-системы	900×900×375 мм
8	Наружный блок сплит-системы	1150×550×1210 мм

Заключение

В ходе выполнения дипломного проекта была разработана программа, которая позволит цифровизировать процесс принятия заявлений на заселение в студенческие общежития, исключая возможность прямого контакта заявителя и сотрудников, что должно ускорить процесс принятия документов и обработки заявлений. В рамках проекта было решено разработать систему для высшего учебного заведения НАО «Алматинский университет энергетики и связи».

При реализации дипломного проекта были выполнены следующие задачи:

- была разработана система приема заявлений на заселение в общежития;
- увеличена скорость обработки заявлений;
- уменьшен путь студента для получения койко-места в студенческом общежитии.

Было произведено экономическое обоснование целесообразности разрабатываемого программного продукта, где было сделано следующее заключение: цена реализации окупает все затраты, потрачены на разработку. Внедрение данного программного продукта в рабочий процесс университета приносит экономию средств и цифровизацию рабочего процесса. Установка данного программного комплекса позволяет сократить штат сотрудников, что в следствии сокращает затраты на выплату налогов, зарплат и прочих денежных расходов. Соответственно, уменьшится объем технического оснащения, амортизационные отчисления и затраты на электроэнергию. Программный комплекс выгодно цифровизирует процесс, облегчив труд персонала и увеличив его КПД, к тому же уменьшив расходы предприятия. Ожидаемый годовой экономический эффект составил 2 200 264 тенге. Приложение окупится в первые 12,8 месяцев использования. Также были исследованы условия труда в помещении, где возможно будет применяться программа, проведен анализ потенциально опасных и вредных факторов в офисе, воздействующих на персонал, и были предложены мероприятия по улучшению качества освещения в помещении, а также был произведен расчет систем кондиционирования.

Список литературы

1 Бардзелл, Джеффри Macromedia Dreamweaver MX 2004 с ASP, ColdFusion и PHP. Из первых рук (+ CD-ROM) / Джеффри Бардзелл. - М.: Эком, 2017. - 560 с.

2 Бенкен, Е. PHP, MySQL, XML. Программирование для Интернета / Е. Бенкен. - М.: БХВ-Петербург, 2016. - 352 с. Wikipedia: Парковочные часы / URL: https://en.wikipedia.org/wiki/Parking_meter

3 Прохоренок, Николай HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера (+ CD-ROM) / Николай Прохоренок. - М.: БХВ-Петербург, 2012. - 912 с.

4 Andy, Harris HTML, XHTML and CSS All-In-One For Dummies® / Andy Harris. - Москва: Наука, 2014. - 173 с.

5 Ben, Henick HTML & CSS – The Good Parts / Ben Henick. - Москва: СИНТЕГ, 2013. - 350 с.

6 Ed, Tittel HTML, XHTML & CSS For Dummies® / Ed Tittel. - Москва: Гостехиздат, 2012. - 416 с.

7 Информационные технологии управления: Учебное пособие/ М.В. Бастриков, О.П. Пономарев; Институт «КВШУ». – Калининград: Изд-во Ин-та «КВШУ», 2005.– 140 с.

8 Информационные технологии управления: Учеб. пособие для вузов/ под ред. проф. Г.А. Титоренко. — 2-е изд., доп. — М.: ЮНИТИ-ДАНА, 2006. - 439 с.

9 Чудновский А.Д. Жукова М.А. Информационные технологии управления в туризме: учебное пособие. - М.:Кнорус,2009 -101 с.

10 Плотникова, Н.И. Комплексная автоматизация туристского бизнеса. Ч.II: Информационные технологии в сфере гостеприимства: Учебно-методическое пособие.-М.: Советский спорт, 2001 -208 с

11 Харрис Энди PHP/MySQL для начинающих; КУДИЦ-Образ - М., 2016. - 384 с.

12 Дронов, В. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов / В. Дронов. - М.: БХВ-Петербург, 2014. - 138 с.

13 Дронов, Владимир HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов / Владимир Дронов. - М.: БХВ-Петербург, 2013. - 416 с.

14 Бенкен, Е. С. PHP, MySQL, XML. Программирование для Интернета (+ CD-ROM) / Е.С. Бенкен. - М.: БХВ-Петербург, 2011. - 304 с.

15 Зандстра Мэтт PHP. Объекты, шаблоны и методики программирования; Вильямс - М., 2016. - 560 с.

16 Колисниченко Денис PHP и MySQL. Разработка Web-приложений; БХВ-Петербург - М., 2017. - 560 с.

17 Квинт, Игорь Создаем сайты с помощью HTML, XHTML и CSS / Игорь Квинт. - М.: Питер, 2014. - 448 с.

18 Основы безопасности жизнедеятельности: Учебное пособие / сост.: М.К.Дюсебаев, Ж.С.Абдимуратов. -Алматы: АУЭС, 2013.-79 с

19 «HTML и CSS. Разработка и дизайн веб-сайтов» /Джон Дакетт; [пер. с англ. М. А. Райтмана]. – Москва: Эксмо, 2019. -480 с.: ил. - (Мировой компьютерный бестселлер)

20 Никсон Робин Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript и CSS; Питер - М., 2017. - 204 с.

21 Справочник экономики URL: <http://ru.solverbook.com/spravochnik/formuly-po-ekonomike/formula-obshhej-rentabelnost/>

22 «JavaScript и jQuery. Интерактивная веб-разработка» / Джон Дакетт; [пер. с англ. М. А. Райтмана]. — Москва: Эксмо, 2017. — 640 с.: ил. — (Мировой компьютерный бестселлер).

23 Документ ГОСТ 12.2.032-78 Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя.

24 Научная интернет библиотека Киберленинка URL: <https://cyberleninka.ru/article/n/sostoyanie-i-tendentsii-razvitiya-rynka-obshchestvennogo-pitaniya-regiona>

25 Научная интернет библиотека Киберленинка URL: <https://cyberleninka.ru/article/n/analiz-rynka-i-klassifikatsiya-predpriyatij-obshchestvennogo-pitaniya>

26 Научная интернет библиотека Киберленинка URL: <https://cyberleninka.ru/article/n/osnovnye-tendentsii-razvitiya-restorannogo-biznesa-v-kazahstane-v-dokrizisnyu-i-krizisnyu-period>

27 Алиев, И. И. Экономика труда: учебник для бакалавров / И. И. Алиев, Н. А. Горелов, Л. О. Ильина. - М.: Юрайт, 2013. - 671 с. - (Серия: Бакалавр. Базовый курс)

28 Мадина Мамырханова, «МОН РК: Места в общежитиях будут предоставляться независимо от формы обучения» URL: <https://kursiv.kz/news/obshchestvo/2018-06/mon-rk-mesta-v-obshchezhitiyakh-budut-predostavlyatsya-nezavisimo-ot-formy?page=59>

29 Александр Мальцев, Маска ввода для HTML элемента input, URL: <https://itchief.ru/lessons/javascript/input-mask-for-html-input-element>

30 Informburo, «КазНУ запустил электронную систему распределения мест в общежитиях», URL: <https://informburo.kz/novosti/kaznu-zapustil-elektronnuyu-sistemu-raspredeleniya-mest-v-obshchezhitiyah.html>

31 Habr, «Обзор Plesk – панели управления хостингом и сайтами», URL: <https://habr.com/ru/company/ruvds/blog/470592/>

Приложение А
Техническое задание

Техническое задание для разработки системы принятия заявлений на заселение в общежитие

Общие требования:

- наименование разрабатываемой системы:
 - системы принятия заявлений на заселение в общежитие.
- цель разработки:
 - реализовать программу для принятия заявлений и цифровизации данных;
 - ускорить процесс принятия документов;
 - дать возможность иногородним студентам отдавать заявления из своих родных городов.
- предлагаемые технологии для разработки программы (на выбор разработчика):
 - jQuery;
 - bootstrap;
 - PHPMyAdmin;
 - Angular;
 - Plesk.
- выбор архитектуры построения:
 - клиент-Сервер.
- предлагаемые языки и технологии программирования:
 - Android;
 - PHP;
 - Java;
 - HTML;
 - CSS;
 - Javascript;
 - C#.
- общий объем программной части системы, Мб
 - не более 1000 Мб.

Технические требования:

- требования к программному обеспечению:

- быстрая скорость обработки входящей информации;

Продолжение приложения А

- формат сайта;
- код программы должен быть читабельным и понятным.
- тестирование и отладка программного обеспечения:
 - проверка работоспособности программы;
 - тестирование платформы на появление ошибок и перегруженности.

Экономические требования:

- расчет стоимости системы и стоимости разработки программного обеспечения (подлежит обсуждению):
 - стоимость готового продукта 2 600 000тг;
 - стоимость разработки 1 800 000тг.
- потенциальные клиенты и области применения:
 - высшие учебные заведения Республики Казахстан.

Приложение Б

Листинг программы

Settings.php

```
<?php
define ('HOST', 'srv-pleskdb33.ps.kz');
define ('USER', 'ugroupk1_base');
define ('PASS', 'Aues2020');
define ('DB', 'ugroupk1_base');
?>
```

Index.php

```
<?php
error_reporting( E_ERROR );
include_once 'setting.php';
session_start();
$CONNECT = mysqli_connect(HOST, USER, PASS, DB);

if ($_SERVER['REQUEST_URI'] == '/') {
    $Page = 'main';
    $Module = 'main';
} else {
    $URL_Path = parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH);
    $URL_Parts = explode('/', trim($URL_Path, '/'));
    $Page = array_shift($URL_Parts);
    $Module = array_shift($URL_Parts);

    if (!empty($Module)) {
        $Param = array();
        for ($i = 0; $i < count($URL_Parts); $i++) {
            $Param[$URL_Parts[$i]] = $URL_Parts[++$i];
        }
        $Module = 'main';
    }

    if (in_array($Page, array('main'))) include("page/$Page.php");
}
```

Продолжение приложения Б

```
else if ($Page == 'proporsals' and in_array($Module, array('add')))  
include("module/proporsals/$Module.php");
```

```
function MessageSend($p1, $p2, $p3 = "", $p4 = 1) {  
    $_SESSION['message'] = '<script type="text/javascript"  
src="/resource/popup.js"></script><div class="popup_notice popup_style_2">  
    <p class="p-modal-main">Благодарим за Ваше заявление</p>  
    <p class="p-modal-content">заявления принимаются с 17 по 21 августа  
</p>  
    <p class="p-modal-content">решение о заселении придет вам на  
указанную в заявлении почту 23 августа 2020 года</p>  
    <p class="p-modal-content">по вопросам обращайтесь по номеру<br>  
+7 (727) 237 89 88  
</p>
```

```
</div>
```

```
'. $p2.'
```

```
</div>';
```

```
if ($p4)
```

```
    Location($p3);
```

```
}
```

```
function MessageShow() {  
if ($_SESSION['message'])$Message = $_SESSION['message'];  
echo $Message;  
$_SESSION['message'] = array();  
}
```

```
function Location ($p1) {  
if (!$p1) $p1 = $_SERVER['HTTP_REFERER'];  
exit(header('Location: '.$p1));  
}
```

```
function FormChars($p1, $p2 = 0) {  
global $CONNECT;  
if ($p2) return mysqli_real_escape_string($CONNECT, $p1);
```

Продолжение приложения Б

```
else return nl2br(htmlspecialchars(trim($p1), ENT_QUOTES), false);
}

function Head($p1) {
echo '<!DOCTYPE html>'
<html>
<head>
<link rel="shortcut icon" href="/resource/favicon-aues-dorm.kz.ico"
type="image/x-icon">
<meta charset="utf-8">
<meta name="keywords" content="" />
<meta name="description" content="">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, user-scalable=no">
<title>'. $p1.' </title>
<link href="/resource/style.css" rel="stylesheet">
<link rel="icon" href="/resource/img/favicon.ico" type="image/x-icon">
<link href="/resource/bootstrap.min.css" rel="stylesheet">
<link rel="stylesheet" type="text/css" href="/resource/main.css">
<script type="text/javascript"
src="/resource/jquery.mobile.custom.min.js"></script>
<script src="/resource/modernizr.custom.js"></script>
<script src="https://code.jquery.com/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstrap.min.js"></script
>

</head>;
}

function ModuleID($p1) {
if ($p1 == 'proporsals') return 1;
else MessageSend(1, 'Модуль не найден.', '/');
}
?>

Add.php
<?php
```

Продолжение приложения Б

```
if ($_POST['enter'] and $_POST['name'] and $_POST['surname'] and
$_POST['patronymic'] and $_POST['email'] and $_POST['phone'] and
$_POST['comment'] and $_POST['iin'] and $_POST['geo'] and $_POST['gender']
and $_POST['nazi'] and $_POST['course'] and $_POST['form_edu'] and
$_POST['stage'] and $_POST['specialty'] and $_POST['privileges']) {
    $_POST['name'] = FormChars($_POST['name']);
    $_POST['surname'] = FormChars($_POST['surname']);
    $_POST['patronymic'] = FormChars($_POST['patronymic']);
    $_POST['email'] = FormChars($_POST['email']);
    $_POST['phone'] = FormChars($_POST['phone']);
    $_POST['comment'] = FormChars($_POST['comment']);
    $_POST['iin'] = FormChars($_POST['iin']);
    $_POST['geo'] = FormChars($_POST['geo']);
    $_POST['nazi'] = FormChars($_POST['nazi']);
    $_POST['gender'] += 0;
    $_POST['course'] += 0;
    $_POST['form_edu'] += 0;
    $_POST['stage'] += 0;
    $_POST['specialty'] += 0;
    $_POST['privileges'] += 0;
```

```
if ($_FILES['file']['tmp_name']) {
    if ($_FILES['file']['type'] != 'application/octet-stream') MessageSend(2, 'Не
верный тип файла.');
```

```
    $_POST['link'] = 0;
    } else $num_file = 0;
```

```
$MaxId = mysqli_fetch_row(mysqli_query($CONNECT, 'SELECT
max(`id`) FROM `proporsals`));
if ($MaxId[0] == 0) mysqli_query($CONNECT, 'ALTER TABLE
`proporsals` AUTO_INCREMENT = 1');
$MaxId[0] += 1;
```

```
if ($_FILES['file']['tmp_name']) {
    foreach(glob('catalog/file/*', GLOB_ONLYDIR) as $num => $Dir) {
        $num_file ++;
        $Count = sizeof(glob($Dir.'/*.*'));
        if ($Count < 250) {
            move_uploaded_file($_FILES['file']['tmp_name'], $Dir.'/'.$MaxId[0].'.zip');
```


Продолжение приложения Б

```
break;
}
}
}
```

```
mysqli_query($CONNECT, "INSERT INTO `proporsals` VALUES
($MaxId[0], '$_POST[name]', '$_POST[surname]', '$_POST[patronymic]',
$_POST[email]', '$_POST[phone]', '$_POST[comment]', '$_POST[iin]',
$_POST[geo]', $_POST[gender]', '$_POST[nazi]', $_POST[course],
$_POST[form_edu], $_POST[stage], $_POST[specialty], $_POST[privileges],
NOW(), $num_file)");
MessageSend(2, '
```

```
    ', '/');
}
Head("") ?>
<body>
<?php
MessageShow()
?>
```

```
Main.php
<?php Head("Заявление на заселение в общежитие НАО "АУЭС");
MessageShow()
?>
```

```
<div class="container-fluid block-add">
  <div class="container block-add-2">
    <div class="row">
      <div class="col-md-1"></div>
      <div class="col-md-10 block">
        <div class="col-md-4 logo">
          <p class="text-center"></p>
        </div>
        <div class="col-md-8 block-content">
          <p class="p-left">Заявление на заселение</p>
          <form method="POST" class="custom_input"
action="/proporsals/add" enctype="multipart/form-data">
            <p>
              <span class="podskazki">Полное ФИО</span> <br>
```

Продолжение приложения Б

```
<input type="text" id="redex" class="form-control
custom_input_input" name="surname" placeholder="Ваша фамилия" required>
<br>
<input type="text" id="redex" class="form-control
custom_input_input" name="name" placeholder="Ваше имя" required> <br>
<input type="text" id="redex" class="form-control
custom_input_input" name="patronymic" placeholder="Ваше отчество" required>
</p>
<p><span class="podskazki">ИИН</span> <br> <input
id="redex" type="text" name="iin" class="custom_input_input form-control iin-
mask" placeholder="____-____-____-____" required> <div class="form-check">
</p>
<p><span class="podskazki">Ваш номер
телефона</span> <br> <input id="redex" type="text" class="custom_input_input
form-control phone-mask" name="phone" placeholder="+7 (____) ____ _ ____"
required></p>
<p><span class="podskazki">Ваш email</span> <br>
<input id="redex" class="custom_input_input form-control " type="email"
name="email" placeholder="info@gmail.com" required></p>
<div class="row">
<div class="col-md-6">
<span class="podskazki">Пол</span> <br>
<select size="1" id="redex" class="form-control"
name="gender" required>
<option value="1">Мужской</option>
<option value="2">Женский</option>
</select>
</div>
<div class="col-md-6">
<span class="podskazki">Ваша
национальность</span> <br>
<input type="text" id="redex" class="form-control"
name="nazi" placeholder="Национальность" required>
</div>
</div>
<p>
<span class="podskazki">Место прописки,
адрес</span> <br>
<span style="float:
left; font-size: 14px; font-weight: light;">укажите через запятую область, город,
район, улицу, дом, квартиру</span>
```

Продолжение приложения Б

```
<input type="text" id="redex" class="form-control
custom_input_input" name="geo" placeholder="область, город, район, улица,
дом, квартира" required>
</p>
<div class="row">
<div class="col-md-4">
<span class="podskazki">Курс</span> <br> <select
size="1" id="redex" class="custom_input_select form-control" name="course"
required>
<option value="1">1</option>
<option value="2">2</option>
<option value="3">3</option>
<option value="4">4</option>
</select>
</div>
<div class="col-md-4">
<span class="podskazki">Форма обучения</span>
<br> <select size="1" class="custom_input_select form-control" id="redex"
name="form_edu" required>
<option value="1">Очное</option>
<option value="2">Заочное</option>
<option value="3">Дистанционное</option>
</select>
</div>
<div class="col-md-4">
<span class="podskazki">Степень</span> <br>
<select size="1" id="redex" class="custom_input_select form-control"
name="stage" required>
<option value="1">Бакалавриат</option>
<option value="2">Магистрант</option>
<option value="3">Докторант</option>
</select>
</div>
</div>
<p style="margin-top: 10px;"><span
class="podskazki">Специальность</span> <br> <select size="1"
class="custom_input_select form-control" id="redex" name="specialty" required>
<option value="1">5B011100 -
Информатика</option>
<option value="2">5B070300 - Информационные
системы</option>
```

Продолжение приложения Б

5B070400 - Вычислительная техника и программное обеспечение

5B100200 - Системы информационной безопасности

5B071900 - Радиотехника, электроника и телекоммуникации

5B071700 - Теплоэнергетика

5B071800 - Электроэнергетика

5B070200 - Автоматизация и управление

5B071600 - Приборостроение

5B074600 - Космическая техника и технологии

5B081200 - Энергообеспечение сельского хозяйства

5B073100 - Безопасность жизнедеятельности и защита окружающей среды

</select></p>

<p>Льготы
<select size="1" id="redex" class="custom_input_select form-control" name="privileges" required>

<input type="radio" value="1"/> Льготы отсутствуют

<input type="radio" value="1"/> Инвалидность

<input type="radio" value="2"/> Многодетная семья (больше четырех детей)

<input type="radio" value="3"/> Сирота

<input type="radio" value="4"/> Утерен кормилец в семье

</select></p>

<p>Загрузите необходимые справки и документы
 все документы загружаются в папку и сюда загружается архив в формате .rar
 <input type="file" class="custom_input_input form-control" name="file"></p>

<p>

Продолжение приложения Б

```
<span class="podskazki">Дополнительный
комментарий/примечание</span>
<textarea type="text" id="redex" name="comment"
class="custom_input_textarea form-control" required>
</textarea>
<br>
<p class="text-center"><input id="ddadd"
class="custom_input_textarea" type="submit" id="inp" class="ct" name="enter"
value="ПОДАТЬ ЗАЯВКУ"></p>
</form>
</div>
</div>
<div class="col-md-1"></div>
</div>
</div>
</div>

<script src="resource/jquery.maskedinput.min.js"
type="text/javascript"></script>
<script src="resource/jquery.maskedinput.js"
type="text/javascript"></script>
<script>
$(document).ready(function() {
$(".phone-mask").click(function(){
$(this).setCursorPosition(3);
}).mask("+7 (999) 999-99-99");
});
</script>
<script>
$(document).ready(function() {
$(".iin-mask").click(function(){
$(this).setCursorPosition(3);
}).mask("999-999-999-999");
});
</script>
</body>
```