

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новгородский государственный университет имени Ярослава Мудрого»
ИНСТИТУТ ЭЛЕКТРОННЫХ И ИНФОРМАЦИОННЫХ СИСТЕМ
ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ

УТВЕРЖДАЮ

Зам. директора по УМ и ВР

_____ Л.Н. Иванова

«___» _____ 2023 г.

РАЗРАБОТКА WEB ПРИЛОЖЕНИЯ ДЛЯ ИЗУЧЕНИЯ АНГЛИЙСКОГО
ЯЗЫКА ПО СРЕДСТВАМ ФЛЭШ КАРТОЧЕК

Пояснительная записка к дипломному проекту по специальности
09.02.03 Программирование в компьютерных системах
ПТК. ДП 9903 11. 000ПЗ

Согласовано:

Консультант по спец. части

_____ Л.Н. Цымбалюк

«___» _____ 2023 года

Консультант по экон. части

_____ Г.В. Лебедева

«___» _____ 2023 года

Нормоконтроль

_____ А.М. Чернега

«___» _____ 2023 года

Заместитель директора по УПР

_____ А.М. Чернега

«___» _____ 2023 года

Руководитель

_____ П.С. Алексеев

«___» _____ 2023 года

Выполнил:

обучающийся группы 9903

_____ Д.П. Казаков

«___» _____ 2023 года

Введение.....	3
1 Общая часть	5
1.1 Постановка задачи.....	5
1.2 Обоснование проектных решения	8
1.3 Обзор и анализ существующих программных систем	9
2 Специальная, практическая, опытно-экспериментальная часть	11
2.1 Анализ задачи	11
2.2 Описание логической структуры.....	13
2.3 Описание работы программы	20
2.4 Руководство оператора	23
3 Экономическая часть	27
3.1 Расчёт основной и дополнительной заработной платы	27
3.2 Расчёт стоимости материалов и лицензионного обеспечения	31
3.3 Расчёт накладных расходов.....	33
3.4 Составление и расчёт цены реализации программного продукта	34
Заключение	35
Список литературы	37
Приложение А	38
Приложение Б	45
Приложение В.....	46
Приложение Г	61
Приложение Д.....	70

Современный мир характеризуется все большей потребностью в знании иностранных языков, особенно английского. Он используется в качестве международного языка коммуникации в бизнесе, научных исследованиях, технологическом развитии и многих других областях. В этой связи создание эффективных и доступных методов обучения английскому языку является актуальной проблемой.

Таким образом, целью данной дипломной работы является разработка веб-приложения для изучения английского языка с помощью флэш-карточек. Предметом исследования, которого является проектирование, разработка и анализ эффективности приложения.

Существует несколько исследований, посвященных разработке веб-приложений для изучения иностранных языков. Однако, большинство из них не используют метод флэш-карточек и не сосредотачиваются на изучении английского языка. Таким образом, данная тема еще не полностью исследована и представляет научный интерес.

Данная дипломная работа предполагает использование методов, включающих разработку веб-приложения, создание базы данных флэш-карточек, проектирование алгоритмов и структур приложения, тестирование и оптимизация, а также анализ эффективности приложения. В результате, мы планируем создать приложение для изучения английского языка, которое будет предоставлять пользователю более продуктивный и интересный способ обучения.

Гипотеза данной работы заключается в том, что использование веб-приложения для изучения английского языка с помощью флэш-карточек будет эффективнее, чем традиционные методы обучения.

Научная новизна работы заключается в том, что она представляет новый подход к изучению английского языка с помощью флэш-карточек в виде веб-приложения. Это приложение будет предоставлять студентам удобный и интерактивный способ изучения английского языка.

Практическая значимость работы заключается в том, что разработанное приложение будет полезно для студентов, изучающих английский язык в учебных заведениях и для людей, изучающих язык самостоятельно. Приложение позволит им улучшить свои знания и навыки в области английского языка.

Для достижения цели работы мы будем использовать методы разработки веб-приложения, такие как язык разметки HTML и языки программирования, CSS, PHP и JavaScript. Также мы будем проводить анализ результатов обучения с помощью статистических методов, чтобы оценить эффективность приложения.

В заключение, данная дипломная работа представляет значимый научный интерес и может привести к созданию более эффективных методов обучения английскому языку. Результаты работы могут быть использованы в учебных заведениях и других образовательных организациях для улучшения качества обучения иностранным языкам.

1 Общая часть

1.1 Постановка задачи

1.1.1 Обоснование необходимости разработки

В соответствии с заданием на выпускную квалификационную работу, необходимо разработать web-приложение по изучению английского языка с помощью флэш-карточек. Данное приложение будет автоматизировать процесс и должно будет содержать в себе следующие функции:

- регистрация и авторизация пользователей в системе;
- создание и управление коллекцией флэш-карточек, содержащих английские слова и их переводы;
- интерактивное изучение английского языка посредством повторения и запоминания английских слов и фраз с помощью флэш-карточек;
- разграничение учётных записей;
- возможность просмотра и редактирования коллекции флэш-карточек, добавление новых слов и переводов;
- предоставление защищённости и хеширования данных пользователя.

Исходя из вышесказанного, построим функциональную модель предметной области информационной системы, рисунок А.1.

Для успешной разработки web-приложения по изучению английского языка необходимо выполнить следующие этапы:

- провести анализ предметной области и изучить существующие аналоги и решения в этой области;
- разработать архитектуру приложения, определить структуру базы данных и логику взаимодействия между компонентами системы;
- реализовать функциональность приложения, включая создание и управление флэш-карточками, систему аутентификации и регистрации пользователей;
- обеспечить безопасность и защиту данных пользователей.

1.1.2 Техничко-математическое описание задачи

Программный продукт будет разработан с помощью применения следующих технологий:

- HTML
- CSS
- JavaScript
- PHP

1.1.2.1 Технология HTML (HyperText Markup Language) является стандартным языком разметки для создания веб-страниц. С помощью HTML можно определить структуру и содержимое страницы, включая текст, изображения, ссылки и другие элементы.

1.1.2.2 Технология CSS (Cascading Style Sheets) является языком таблиц стилей, который позволяет определять внешний вид и оформление веб-страниц. С помощью CSS можно задавать цвета, шрифты, размеры и расположение элементов на странице, создавая привлекательный и современный дизайн.

1.1.2.3 Технология JavaScript является скриптовым языком программирования, который позволяет добавлять интерактивность и динамическое поведение на веб-страницах. С помощью JavaScript можно создавать функции, обрабатывать события, выполнять проверки данных и взаимодействовать с пользователем.

1.1.2.4 Технология PHP (Hypertext Preprocessor) является серверным скриптовым языком программирования, предназначенным для разработки веб-приложений. С помощью PHP можно обрабатывать данные на сервере, взаимодействовать с базами данных и создавать динамические веб-страницы.

1.1.3 Характеристика бизнес-процессов

В рамках дипломного проекта разрабатывается web-приложение и для успешного функционирования приложения требуется создание базы данных, которая будет содержать информацию о пользователях системы и другие необходимые данные. Пользователей необходимо классифицировать по уровням доступа к системе, для этого необходимо определить основные функции, которые будут выполняться разными категориями пользователей.

Для более наглядного представления предметной области и категорий пользователей, мы строим организационную диаграмму предметной области, рисунок А.2. Пользователи разрабатываемой системы будут разделены на две группы: пользователи с неограниченным доступом и пользователи с ограниченным доступом, рисунок А.3.

Для визуализации бизнес-процессов проектной деятельности мы создаем диаграмму бизнес-прецедентов, рисунок А.4. На данной диаграмме можно наглядно увидеть, какие функции могут выполнять пользователи разрабатываемой системы в рамках бизнес-процессов.

1.1.4 Требования к оборудованию

Для приложения, разработанного на базе HTML, CSS, JS и PHP, рекомендуется следующее оборудование:

Компьютер или ноутбук с процессором x86 (Intel, AMD) и тактовой частотой не менее 1,8 ГГц. Рекомендуется использовать процессор с двумя или более ядрами для повышения производительности.

Оперативная память объемом не менее 2 ГБ. Рекомендуется иметь 4 ГБ ОЗУ или больше для более плавной работы приложения. Минимальный объем памяти для выполнения на виртуальной машине составляет 2,5 ГБ.

Доступное место на жестком диске от 2 до 10 ГБ для установки и хранения приложения и соответствующих данных.

Жесткий диск со скоростью вращения 5400 об/мин или выше обеспечит более быстрый доступ к данным и улучшит производительность работы приложения.

Видеоадаптер, поддерживающий DirectX 9 и способный отображать разрешение экрана не менее 1024x768 пикселей, обеспечит качественное визуальное представление флэш-карточек и других элементов приложения.

Операционная система: такие как Windows, macOS, Linux, рекомендуется использовать последние версии операционных систем, чтобы обеспечить доступ к последним функциям и обновлениям.

Браузер: Google Chrome, Mozilla Firefox, Safari, Opera и т.д. рекомендуется использовать последние версии браузеров для обеспечения лучшей производительности, безопасности и совместимости с новыми возможностями языков и технологий.

Интернет-соединение: для полноценного использования web-приложения необходимо иметь стабильное и достаточно быстрое интернет-соединение. Широкополосное соединение, такое как высокоскоростной доступ к интернету через проводную сеть или беспроводную Wi-Fi, рекомендуется для оптимальной загрузки и обмена данными с сервером приложения.

1.2 Обоснование проектных решения

1.2.1 Обоснование выбора языков программирования

Выбор данных инструментов и языков программирования (HTML, CSS, JS, PHP) обусловлен их широкой распространенностью, поддержкой соответствующими технологиями и наличием богатого набора инструментов и ресурсов для разработки веб-приложений.

1.2.2 Инструментальные средства

Для хранения и управления данными, связанными с приложением, будет использоваться HeidiSQL. HeidiSQL - это бесплатный и открытый инструмент для работы с базами данных MySQL, MariaDB, Microsoft SQL Server и PostgreSQL. С помощью HeidiSQL разработчики могут удобно создавать и изменять структуру базы данных, выполнять запросы, управлять данными и обеспечивать безопасность информации. Этот инструмент предоставляет

графический интерфейс, который облегчает работу с базами данных и упрощает выполнение административных задач.

HeidiSQL предоставляет удобные инструменты для работы с базой данных, что позволяет эффективно управлять информацией, связанной с приложением.

1.2.3 Обоснование выбора среды программирования

В качестве среды программирования для разработки web-приложения будет использоваться Visual Studio Code. Visual Studio Code является популярным и мощным инструментом разработки, предоставляющим широкие возможности для написания кода на различных языках программирования, включая HTML, CSS, JS и PHP. Она обладает удобным пользовательским интерфейсом, поддерживает авто дополнение кода, отладку и интеграцию с различными расширениями для облегчения разработки и повышения производительности.

1.2.4 Информационное обеспечение

Для построения диаграмм и блок-схем был выбран редактор - MS Visio 2016. Для оформления документации к приложения был выбран текстовый редактор - MS Word. Для презентации работы была выбрана программа для подготовки презентаций и их просмотра - MS PowerPoint. Все эти программы являются частью офисного пакета приложений - Microsoft Office. И их интегрированность друг с другом позволяет существенно упростить работу.

1.3 Обзор и анализ существующих программных систем

Были рассмотрены некоторые аналоги уже существующих приложений. Вот некоторые из них и их плюсы и минусы:

- duocards;
- memrise.

1.3.1 Duocards

Duocard - это приложение для изучения иностранных языков с помощью карточек. Duocard использует метод обучения с помощью карточек и пространственного повторения, что позволяет легко и быстро усваивать новые слова и фразы.

Преимущества Duocards:

- большое количество поддерживаемых языков;
- видеоуроки для изучения языков;
- наличие приложений под платформы Android и IOS;
- предусмотрена функция аудио воспроизведения карточек;
- возможность найти значения карточек через ИИ.

Недостатки Duocards:

- ограничений количества использований в бесплатной версии;
- отсутствие готовых наборов карточек для изучения;
- предусмотрена функция аудио воспроизведения карточек;
- не корректно работающий интерфейс в браузерной версии приложения.

1.3.2 Memrise

Memrise это приложение для изучения иностранных языков, которое предлагает множество функций и возможностей.

Преимущества Memrise:

- ежедневные интерактивные бонусы, побуждающие на ежедневное изучение языка;
- дружелюбный интерфейс;

Недостатки Memrise:

- ошибки в переводах;
- многие функции доступны только в платной версии;

2 Специальная, практическая, опытно-экспериментальная часть

2.1 Анализ задачи

2.1.1 Информационное моделирование предметной области

На основе анализа предметной области при моделировании базы данных были выделены следующие сущности:

- words (слова), атрибуты: id (уникальный идентификатор), word (слово), definition (определение/перевод);
- user (пользователи), атрибуты: id (уникальный идентификатор), login (логин), pass (пароль), email (электронная почта);

ER-модель представлена на рисунке А.5 и реляционная модель данных, представленная на рисунке А.6.

Произведем анализ сущностей:

- сущности находятся в первой нормальной форме, так как удовлетворяют следующему требованию: сущность не должна иметь многозначных атрибутов;
- сущность находится во второй и третьей нормальной форме, так как она удовлетворяет условиям что ни одно из неключевых полей таблицы не идентифицируется с помощью другого неключевого поля и нет составных ключей.

В СУБД для защиты базы данных были реализованы следующие методы защиты:

- защита паролем, шифрование данных (алгоритм хэширования MD5);
- разграничение прав доступа к объектам базы данных;
- поддержание целостности данных в СУБД.

2.1.2 Проектирование пользовательского интерфейса

На основе эскизов были созданы формы:

- экран регистрации пользователя, рисунок Г.1;
- экран авторизации пользователя, рисунок Г.2;
- экран главного меню пользователя, рисунок Г.3;
- экран просмотра и редактирования словаря, рисунок Г.4;
- экран просмотра информации о проекте, рисунок Г.5;
- экран меню выбора набора карточек, рисунок Г.6;
- экран изучения флэш-карточек, рисунок Г.7.

Описание разработанного интерфейса:

- экран регистрации пользователя используется для регистрации аккаунта нового пользователя в системе, рисунок Г.1;
- экран авторизации пользователя используется для входа и дальнейшего использования приложения, рисунок Г.2;
- экран главного меню пользователя используется для навигации по приложению, рисунок Г.3;
- экран просмотра и редактирования словаря пользователя используется для работы пользователя с личным словарем данных, рисунок Г.4;
- экран просмотра информации о проекте используется для ознакомления с информацией, услугами и контактными данными проекта, рисунок Г.5;
- экран меню выбора набора карточек используется для выбора пользователем необходимого ему набора карточек для дальнейшего изучения, рисунок Г.6;
- экран изучения флэш-карточек используется пользователем для изучения и практики слов английского языка, рисунок Г.7.

2.2 Описание логической структуры

2.2.1 Алгоритм программы

На форме регистрации пользователь имеет возможность зарегистрировать новый аккаунт в приложении, а также ознакомиться со всей информацией о проекте по средствам вкладок навигационного меню. Для заполнения формы регистрации пользователю необходимо придумать и заполнить текстовые поля, содержащие логин, электронную почту и пароль. После заполнения данные проходят валидацию. Длина логина и пароля должна быть не менее трёх символов, длина электронной почты не менее пяти символов и содержать символ '@'. Так же проходит проверка логина на уникальность и в случае наличия аккаунта с подобным логином в приложении выдается ошибка "Пользователь с таким логином уже существует". Если все обязательные поля были заполнены, то программа зашифрует пароль пользователя с помощью технологии шифрования MD5 для более надёжного хранения его в базе данных, далее она выполнит запрос на добавления всех данных в таблицу «User» (Пользователи) базы данных, а также создаст таблицу «Words_Имя пользователя» для дальнейшего использования для личного словаря каждого пользователя. После этого пользователь будет перенаправлен на форму авторизации. Интерфейс формы регистрации представлен рисунке Г.1.

При запуске web-приложения открывается страница сайта с формой авторизации в системе приложения. На странице авторизации пользователь может войти в аккаунт приложения для дальнейшего использования функциональных возможностей приложения, а также ознакомиться со всей информацией о проекте по средствам вкладок навигационного меню. При вводе логина и пароля в соответствующие поля и нажатии кнопки авторизации, программа проверит наличие введенных данных в базе данных с помощью запроса к базе данных, заранее программа зашифрует пароль, введенный пользователем, с помощью технологии шифрования MD5, так как именно в таком виде он хранится в базе данных. Если данные введены некорректно, то

программа выдаст сообщение об ошибке с текстом «Имя пользователя, либо пароль введены не верно!», если данные введены корректно, то приложение успешно авторизует пользователя и перенаправит его на страницу главного меню, для дальнейших действий. Также, если пользователь не зарегистрирован в приложении под формой меню авторизации расположена гиперссылка для регистрации. По нажатию которой форма авторизации сменяется на форму регистрацию. Интерфейс формы авторизации представлен на рисунке Г.2.

На форме главного меню представлено следующее: Навигационное меню с кнопками «Мой словарь», которая переносит пользователя в редактирование личного набора карточек, кнопка «Флэш карточки» служащая для перехода в меню выбора набора карточек для изучения. Также в верхней части пользователя встречает приветственная надпись с его логином и кнопкой «Выйти» для выхода или смены аккаунта. На каждой странице веб-приложения, в верхнем навигационном меню «header» присутствует переключатель языка. С помощью которого пользователь может сменить на каком языке будет отображаться интерфейс web приложения. Интерфейс формы главного меню пользователя представлен на рисунке Г.3.

На форме просмотра и редактирования пользовательского словаря пользователю предоставляются слова из его личного набора карточек для изучения. Слова представлены в виде таблицы, состоящей из двух столбцов, термина на английском языке и его перевода на русский язык. Предусмотрена возможность добавления новых слов, а также редактирования или удаления уже существующих. Интерфейс формы просмотра и редактирования пользовательского словаря представлен на рисунке Г.4.

На форме просмотра информации о проекте пользователь может ознакомиться с основной информацией о проекте, которая в себя включает: краткое описание проекта, перечень предоставляемых услуг и контактную информацию для связи с командой проекта. Предусмотрена отдельно разработанная модель верхнего навигационного меню, которое появляется сверху в независимости от того насколько вниз пользователь пролистал сайт, и

имеет активные ссылки-якоря для перехода на нужную пользователю часть информации о проекте. Это способствует удобному ознакомлению с представленной информацией и навигации по ней. Помимо этого, в навигационное меню встроена кнопка «Начать!», которая ведёт на страницу главного меню или же на страницу авторизации в зависимости от того, вошёл ли пользователь в аккаунт. Интерфейс формы просмотра информации о проекте представлен на рисунке Г.5.

На форме меню выбора набора карточек перед пользователем представлен выбор из трёх различных наборов карточек с их описанием. «Наиболее распространённые слова», который включает в себя самые распространённые и часто используемые слова на английском языке, «Сленговые карточки» содержащий набор сленговых слов или выражений в английском языке и «Персональные карточки» содержащий в себе набор личных карточек пользователя, созданных для изучения. Меню выбора представлено в приятном минималистичном стиле с добавленными анимационными эффектами при наведении на наборы. Интерфейс формы меню выбора набора карточек представлен на рисунке Г.6.

На форме приложения флэш-карточек представлена основная функция web приложения, а именно флэш карточки по изучению английского языка. По центру экрана представлен стилизованный контейнер в виде карточки. На передней стороне карточек располагается слово или словосочетание на английском языке. По нажатию на саму карточку происходит анимация переворота и на обратной стороне карточки пользователь видит перед собой перевод данной карточки на русский язык и изначальный вариант термина на английском языке, обозначенный второстепенным шрифтом. Вернуть карточку в исходное положение можно просто убрав с неё курсор мыши, либо при нажатии кнопки «Следующее слово» в нижней панели кнопок. Нижняя функциональная панель кнопок видоизменяется в зависимости от прав пользователя. Для обычного пользователя функции добавления, удаления или редактирования доступны только в наборе «Персональные карточки». Нижняя

функциональная панель кнопок представляет из себя набор из четырёх кнопок: кнопка «Следующее слово» меняет текущее слово на другое слово из текущего набора в случайно сгенерированном порядке, кнопка в виде иконки с мусорным баком служит для удаления из набора карточек текущей карточки и по нажатию вызывает модальное окно, в котором пользователь имеет возможность подтвердить или отменить удаление текущего слова. Кнопка, представляющая из себя иконку с бланком и карандашом служит для редактирования текущего слова, вызывает модальное окно, в котором пользователь может ввести новое значение термина или перевода для текущего слова. И кнопка, представляющая из себя иконку со знаком плюс обведённую в кружок, служащая для добавления нового слова в набор карточек. Она открывает модальное окно, где пользователь может ввести термин и перевод нового слова, которое будет добавлено в набор. В левой части формы представлен заголовок текущего набора карточек и количеством карточек в данном наборе. Интерфейс формы приложения флэш-карточек представлен на рисунке Г.7.

2.2.2 Используемые методы

Для обеспечения функциональности web приложения в процессе разработки в структуре проекта была определена отдельная папка «actions», в которой были в последствии созданы следующие методы:

- addWord.php;
- auth.php;
- editWord.php;
- exit.php;
- reg.php;
- updateWords_common.php;
- updateWords_slang.php;
- updateWords.php;
- wordDel.php.

Рассмотрим метод `reg.php`. Данный метод служит для регистрации нового аккаунта в системе. Прежде всего происходит обработка переданных значений из формы регистрации. Применяется встроенная функция `trim()` для удаления лишних пробелов в начале и конце строки. И затем применяется функция `filter_var()` с параметром `FILTER_SANITIZE_STRING` для фильтрации и очистки значения, чтобы оно содержало только допустимые символы строки, игнорируя или удаляя любые потенциально опасные или нежелательные символы. В продолжении этой строки проходят валидацию. Логин должен быть не менее трёх символов в длину и иметь уникальное значение. Если логин не соответствует условию по допустимой длине будет выведена ошибка о недопустимой длине. Проверка на уникальность логина происходит по средствам SQL запроса на наличие существующего пользователя с таким логином в базе данных. При несоблюдении условий уникальности выдастся ошибка что пользователь с таким логином уже существует. Пароль должен состоять не менее чем из трёх символов в длину, в противном случае будет выдана ошибка о недопустимой длине пароля. При проверке адреса электронной почты, требуется, чтобы длина адреса составляла не менее пяти символов и включала символ "@". В случае несоблюдения этих условий, будет выведено сообщение об ошибке, указывающее на недопустимую длину адреса электронной почты или на необходимость наличия символа "@" в электронной почте. После этого происходит хэширование пароля по средствам функции `md5()`, подключение к базе данных и внесение данных о новом пользователе и добавление таблицы с персональными карточками пользователя.

Код метода `reg.php` представлен в приложении В, листинг В.1.

Рассмотрим метод `auth.php`. Данный метод служит для аутентификации пользователя в системе. В начале переданные данные логина и пароля из HTTP POST запроса проходят фильтрацию и очистку от нежелательных символов с использованием функции `filter_var()`, после переданный пароль шифруется и происходит подключение к базе данных. По средствам SQL выборки проверяется существует ли указанный пользователь и совпадает ли введённый

пароль. В случае несовпадения одного из пунктов будет выведено сообщение об ошибке о том, что имя пользователя либо пароль введены не верно. Если пользователь найден в базе данных, устанавливается cookie «user» с именем пользователя и сроком действия в 1 час. Происходит закрытие соединения с базой данных, и пользователь перенаправляется в главное меню web приложения.

Код метода `auth.php` представлен в приложении В, листинг В.2.

Рассмотрим метод `exit.php`. Данный метод служит для выхода пользователя из аккаунта. При вызове данного метода из главного меню по кнопке «Выйти» происходит удаление cookie и пользователь перенаправляется в главное меню, которое в свою очередь совершит переадресацию на страницу аутентификации так как пользователь не аутентифицирован в системе.

Код метода `exit.php` представлен в приложении В, листинг В.3.

Рассмотрим метод `addWord.php`. Данный метод используется для добавления новой карточки в персональный словарь пользователя. Входные данные термина и перевода проверяются на заполненность с помощью функции `empty()`, если одно из значений не заполнено возникнет ошибка, сообщающая что не заполнено одно из значений. Если значения введены корректно, то будет произведено подключение к базе данных и внесено новое определение в персональный словарь пользователя. В зависимости от того какая страница вела на метод добавления слова пользователь будет перенаправлен на предыдущую страницу по средствам исполнения JavaScript с функцией `document.referrer.indexOf` так как добавление нового слова может происходить из формы просмотра словаря, так и из самого приложения флэш карточек.

Код метода `addWord.php` представлен в приложении В, листинг В.4.

Рассмотрим метод `wordDel.php`. Данный метод служит для удаления слова из персонального словаря пользователя. Передаёт данные необходимого для удаления слова через HTTP POST запрос и с помощью SQL запроса удаляет данное слово из таблицы. После выполнения метода идёт переадресация на страницу вызова метода.

Код метода `wordDel.php` представлен в приложении В, листинг В.5.

Рассмотрим метод `editWord.php`. Данный метод служит для редактирования данных флэш карточки. Введённые данные для изменения пользователем в модальное окно по средствам HTTP POST запроса передаются в метод. Происходит очистка входных данных на нежелательные символы и по средствам SQL запроса происходит изменение данных в базе.

Код метода `editWord.php` представлен в приложении В, листинг В.6.

Рассмотрим метод `updateWords.php`. Данный метод необходим для обновления текущей флэш карточки без необходимости обновлять страницу. Выполняя подключение к базе данных выбирает случайное значение из набора карточек и выводит данные в формате JSON. С этим методом не отдельно связана функция JavaScript `updateWords()`. Которая в свою очередь служит для обновления элементов на web странице с помощью полученных данных. Методы `updateWords_common.php` и `updateWords_slang.php` полностью идентичны методу `updateWords.php` за исключением того, что служат для обновления текущих флэш карточек в других наборах карточек.

Код метода `updateWords.php` представлен в приложении В, листинг В.7.

2.2.3 Составные части программы и связи между ними

Сайт можно разделить на несколько страниц:

- страница регистрации-авторизации;
- главная страница;
- страница редактирования личного словаря;
- страница с информацией проекта;
- страница изучения флэш карточек.

Каждая страница имеет свой набор функций, для которых используются вспомогательные файлы с функциями.

2.2.4 Структура проекта

Сайт состоит из разных файлов, отвечающих за различный функционал на сайте. Файлы размещены в разных папках в соответствии с их назначением. Структура проекта представлена в таблице 2.

Таблица 1 – Структура проекта

Название папки	Описание	Содержимое
localhost	Папка с основными блоками сайта	index.php – файл регистрации-аутентификации
		core.php – файл главного меню
		info.php – файл с информацией проекта
		words.php – файл с личным словарём пользователя
		selector.php – файл выбора набора карточек
		app.php – файл приложения флэш карточек
css	Папка с файлами стилей сайта	styles.css – файл стилей для регистрации-авторизации
		core.css – файл стилей для главного меню
		info.css – файл стилей для страницы информации
		words.css – файл стилей для страницы словаря пользователя
		selector.css – файл стилей для страницы выбора набора карточек
		app.css – файл стилей для приложения флэш карточек
img	Папка для хранения загружаемых обложек альбомов	Изображения
js	Папка с файлами JavaScript	script.js – файл с функциями для авторизации-регистрации
		app.js – файл с функциями для приложения флэш карточек
		core.js – файл с функциями для главного меню
		info.js – файл с функциями для страницы информации
		selector.js – файл с функциями для меню выбора набора карточек
		words.js – файл с функциями для страницы редактирования карточек
ru	Папка с блоками сайта на русском языке	Блоки сайта в переводе аналогичные папке localhost

2.3 Описание работы программы

2.3.1 Общие сведения

Разрабатываемый проект представляет собой сайт, при разработке которого языки PHP, JavaScript, HTML, CSS, SQL. Также использовались такие средства разработки как:

- локальный сервер Open Server Panel;
- редактор Visual Studio Code;
- приложение для работы с базой данных HeidiSQL.

2.3.2 Функциональное назначение

Функциональным назначением web приложения является предоставление возможности бесплатного, интерактивного и практичного инструмента по изучению английского языка по средствам флэш карточек, с возможностью настройки личного словаря с флэш карточками и его всевозможного редактирования.

2.3.3 Входные данные

Входными данными являются:

- логин;
- пароль;
- электронная почта;
- термины флэш карточек;
- переводы флэш карточек;

2.3.4 Выходные данные

Выходными данными являются:

- логин;
- термины на флэш карточках;
- переводы на флэш карточках;
- количество флэш карточек;
- название выбранного набора карточек;
- прочая информация сайта.

2.3.5 Тестирование

Тестирование позволяет заранее узнать, как поведет себя программа в определенной ситуации до того, как она попадет к конечному пользователю. Это крайне важно, особенно в данном случае, так как код сайта постоянно меняется в течении разработки и не всегда возможно обнаружить ошибки в коде и исправить их.

Первым параметром является отображение сайта в различных браузерах. Для этого необходимо запустить сайт на различных браузерах и проверить отображение основных элементов интерфейса.

Как видно из рисунков Г.11, Г.12 и Г.13 – сайт корректно отображается на трех самых распространенных браузерах Google Chrome, Yandex Browser и Microsoft Edge.

Следующим параметром будет проверка добавления записей в базу данных.

На рисунке Г.14 представлен процесс добавления новой флэш карточки в базу данных.

На рисунке Г.15 видно, что новая запись была успешно внесена в базу данных.

В течение тестирования итоговый результат совпал с ожидаемым, что подтверждает верность логики работы программы.

2.3.6 Вызов и загрузка

В процессе разработки открытие сайта осуществляется путём запуска проекта в локальном веб - сервере Open Server Panel, подключение и загрузка базы данных происходит автоматически. После возможна полноценная работа с сайтом.

По окончании разработки необходимо загрузить файлы сайта на сервер и присвоить сайту доменное имя. Загрузка файлов осуществляется с помощью FTP – клиента или встроенного на хостинге файлового менеджера.

После размещения сайта на сервере он будет доступен в сети Интернет по присвоенному доменному имени.

2.4 Руководство оператора

2.4.1 Назначение программы

Web приложение предназначено для удобного изучения английского языка по средствам флэш карточек с прилагающейся функциональной средой.

2.4.2 Условия выполнения

Для работы с сайтом пользователю требуется иметь смартфон, планшет или персональный компьютер с Интернет - соединением, а также должен быть установлен любой веб-браузер.

2.4.3 Выполнение программы

2.4.3.1 Работа пользователя

Пользователь может изучать готовые наборы флэш карточек, а также создать свой персональный набор карточек с предусмотренными функциями добавления, удаления и изменения карточек, а также просматривать всю доступную информацию о проекте и т.д.

При открытии сайта пользователь попадет на страницу регистрации-аутентификации сайта. Рисунок Г.1.

Если у пользователя уже есть аккаунт он может нажать на гиперссылку «Войти», и страница регистрации сменит форму на меню аутентификации пользователя. Рисунок Г.2.

При успешной аутентификации пользователь попадает на страницу главного меню, которое служит для навигации в приложении. Сверху отобразится логин пользователя и приветственная надпись. А также кнопка «Выйти» для выхода из аккаунта. Рисунок Г.3.

При нажатии кнопки «Мой словарь» произойдёт переход на страницу просмотра и редактирования личного словаря пользователя. Рисунок Г.4.

При нажатии на одну из кнопок верхнего навигационного меню произойдёт переход на информационную страницу о проекте. Рисунок Г.5.

При нажатии на кнопку «Флэш карточки» пользователя перенаправит на страницу меню выбора набора флэш карточек для изучения. Рисунок Г.6.

При дальнейшем переходе на один из наборов флэш карточек откроется само приложение для изучения флэш карточек. Пользователю будет представлен стилизованный под карточку контейнер. По нажатию на карточку она разворачивается на обратную сторону, где пользователь может увидеть перевод и исходный термин на английском языке. Рисунок Г.7.

По нажатию на кнопку «Следующее слово» текущее слово будет обновлено на иное случайное слово из выбранного набора флэш карточек.

По нажатию на кнопку с плюсом, обведённым в кружке на странице приложения или же кнопку «Добавить» из окна просмотра и редактирования персонального набора карточек для пользователя будет выведено модальное окно добавления с помощью, которого он сможет ввести и добавить новый термин на английском языке и перевод к нему. Рисунок Г.8.

При нажатии на кнопку с эмблемой мусорной корзины на странице приложения или же кнопку «Удалить» из окна просмотра и редактирования персонального набора карточек для пользователя будет выведено модальное окно удаления слова, с помощью которого он сможет удалить текущее либо выделенное слово из базы данных. Рисунок Г.9.

При нажатии на кнопку с эмблемой блокнота и карандаша на странице приложения или же кнопку «Редактировать» из окна просмотра и редактирования персонального набора карточек для пользователя будет выведено модальное окно редактирования слова, с помощью которого он сможет изменить текущее либо выделенное слово изменив термин на английском языке либо его перевод. Рисунок Г.10.

2.4.3.2 Техника безопасности при работе с компьютером

Пользователь обязан:

- знать правила технической эксплуатации оборудования, предоставленного для работы на персональном компьютере;
- содержать в чистоте рабочее место;

- соблюдать правила гигиены;
- знать местонахождение и уметь использовать средства индивидуальной защиты и оказания первой медицинской помощи;
- сообщать руководителю о всех случаях травмирования, плохого самочувствия пользователей, неисправности оборудования.

Пользователю запрещается:

- производить самостоятельное вскрытие и ремонт электрооборудования, бытовых приборов, компьютеров и копировально-множительной техники;
- работать на компьютере при снятых защитных кожухах или при закрытых вентиляционных отверстиях кожуха;
- прикасаться к задней панели компьютеров, копировально-множительной техники, другого оборудования и приборов при включенном питании;
- переключать разъемы кабелей компьютеров, копировально-множительной техники, другого оборудования и приборов при включенном питании.
- отключать оборудование от электросети выдергиванием электровилки, держа за шнур;
- пользоваться неисправными выключателями, розетками, компьютерами, копировально-множительной техникой, другим оборудованием и приборами;
- подключать к одному источнику тока более двух потребителей тока или хотя бы один потребитель, мощность потребления тока которого превышает мощность источника тока;
- дотрагиваться до кабеля питания мокрыми руками, ставить на него тяжелые предметы;
- устанавливать на оборудование емкости с жидкостью, тяжелые предметы, мелкие металлические предметы. Попадание внутрь влаги и металлических предметов может вызвать замыкание и пожар;
- подвергать оборудование ударам и другим механическим воздействиям;

- работать при закрытых вентиляционных отверстиях оборудования;
- запрещается курение в помещениях, коридорах и лестничных переходах, курение разрешается только в специально отведенных для этой цели местах.

Начиная с третьего часа работы, проводить через каждые 50 минут десятиминутные регламентированные перерывы. Продолжительность непрерывной работы с компьютером не должна быть более 2-х часов.

Через каждые 20 минут работы с компьютером необходимо отводить взгляд от экрана монитора и выполнять расслабляющие упражнения для глаз.

Во время перерывов, с целью снижения нервно-эмоционального напряжения, утомления зрительного анализатора, устранения влияния гиподинамии и утомления, выполнять комплексы физических упражнений.

Во всех случаях прекращения работы оборудования, обрыва проводов питания, неисправности заземления, искрении, появления характерного запаха гари, немедленно отключить питание и сообщить об аварийной ситуации руководителю.

3 Экономическая часть

3.1 Расчёт основной и дополнительной заработной платы

Смета затрат на производство – это общий свод плановых затрат в стоимостном выражении на производство продукции, выполнение работ и услуг в соответствии с производственной программой предприятия. Смета затрат на производство разрабатывается по единой номенклатуре однородных экономических элементов.

Состав затрат предприятия конкретной отрасли определяется в соответствии с Типовыми рекомендациями по учету и подсчету себестоимости в промышленности, сельском хозяйстве, строительстве, торговле, общественном питании и в других отраслях, разрабатываемыми министерствами, ведомствами по согласованию с Минфином РФ. В состав затрат включают расходы на:

- производство продукции, то есть выполнение работ, оказание услуг;
- управление и сбыт продукции, не входят в состав затрат на производство

расходы по капитальным вложениям, на освоение новых видов продукции, непроизводственные расходы.

Группировка затрат в смете по экономическим элементам:

- материальные затраты, такие как сырье и материалы, покупные комплектующие изделия, полуфабрикаты, топливо, запасные части и пр.;
- затраты на оплату труда, такие как заработная плата, премии за достижение производственных показателей, выплаты в виде выслуги лет, резерв на оплату отпусков;
- отчисления во внебюджетные фонды;
- амортизационные отчисления;
- прочие затраты, такие как оплата услуг сторонних организаций, хозяйственные расходы, расходы по рекламе и сбыту продукции и др.

Расчет сметы затрат на производство осуществляется на основе бизнес-плана в части производственной программы на будущий финансовый год, а

также анализа фактических данных прошедшего периода и расчетных нормативов по статьям затрат, принятых в организации.

Расчетные нормативы по статьям затрат планируются на основе анализа действующих рыночных цен на аналогичную выпускаемую продукцию, фактических цен реализации продукции прошедшего финансового года.

По этим данным определяется предельный норматив затрат на производство при существующем уровне рыночной, договорной цены. Рассчитанный таким образом предельный норматив затрат на единицу продукции с учетом принятых на предприятии нормативов по статьям затрат позволяет определить плановую смету в разрезе статей и элементов.

С учетом специфики отраслевой деятельности организации составление сметы затрат на производство ведется по сегментам или видам деятельности, видам продукции, работ, услуг; дальнейшая группировка возможна по структурным подразделениям.

На заключительном этапе планирования сметы затрат на производство составляется сводная смета под плановую производственную программу с учетом объемов выпуска.

На первом этапе разработки программного продукта необходимо рассчитать смету затрат, а также определить этапы разработки web приложения для изучения английского языка при помощи флэш карточек и трудоёмкость выполнения каждого этапа. Все показатели представлены в таблице 2.

Таблица 2 — Этапы и трудоёмкость выполнения работ в часах

Этапы разработки	Руководитель	Техник
Постановка задачи	1	5
Выбор инструментальных средств	1	4
Разработка мат. модели	1	14
Построение алгоритма	2	16
Программирование	1	52
Тестирование	1	25
Отладка	1	30
Написание пояснительной записки	2	30
Итого	10	176

Минимальный размер заработной платы на предприятии не может быть ниже минимального размера заработной платы, установленного правительством.

Предполагаем, что месячный оклад руководителя будет составлять 27000 руб., а техника 16242 руб.

3.1.1 Расчёт основной заработной платы

При расчёте основной заработной платы учитывается заработная плата всех категорий работников, непосредственно занятых разработкой программного продукта. Размер заработной платы определяется исходя из количества исполнителей и их квалификационного уровня, а также затраченного ими времени в целом на разработку программного продукта и отдельных его этапов.

Основная заработная плата работников определяется по повременной системе оплаты труда.

Рассчитаем цену одного часа работы сотрудника по формуле 1:

$$\text{Цч} = \text{Ом} / (\text{Др} \times \text{Ч}), \quad (1)$$

где Др — среднее число рабочих дней в месяце;

Ч — часов в смене.

Произведем расчёт оплаты труда за час исходя из 8-ми часовой и 5-ти дневной рабочей недели.

Руководитель проекта:

$$\text{Ц}_ч = 27000 / (20 \times 8), = 168,75 \text{ руб.}$$

Техник:

$$\text{Ц}_ч = 16000 / (20 \times 8), = 101,51 \text{ руб.}$$

Заработная плата работников по окладу рассчитывается по формуле 2

$$\text{ЗП}_{\text{осн}} = \text{Ц}_ч \times \text{Ч}, \quad (2)$$

где Ч — количество часов, затраченных на разработку программного продукта.

Таблица 3 — Основная заработная плата работников в рублях

Должность	Количество часов, затраченных на разработку продукта, час	Оплата труда за 1 час	Заработная плата
Руководитель проекта	14	168,75	2362,5
Техник	176	100	17865,7
Итого			20228,2

Таким образом, по показателям, представленным в таблице 3, основная заработная плата работников составила 20228,2 руб.

3.1.2 Расчет дополнительной заработной платы

Дополнительная заработная плата включает различные виды доплат сверх основной заработной платы: премии, доплату за работу в сверхурочное время, надбавки за профессиональное мастерство, оплату очередного и учебных отпусков и прочие виды доплат.

Дополнительная заработная плата устанавливается на предприятии в процентах от суммы основной заработной платы и составляет от 16 до 20%.

Таким образом, дополнительная заработная плата будет определяться по следующей формуле:

$$\text{ЗП}_{\text{доп}} = \text{ЗП}_{\text{осн}} \times 20 / 100 \quad (3)$$

где $\text{ЗП}_{\text{осн}}$ — основная заработная плата, руб.

Значение $\text{ЗП}_{\text{осн}}$ возьмём из таблицы 3, и рассчитаем дополнительную заработную плату.

$$ЗП_{\text{доп}} = 20228,2 \times 0,2 = 4045,6 \text{ руб.}$$

Теперь получим сумму основной и дополнительной заработной платы, $ЗП_{\text{сум}}$:

$$ЗП_{\text{сум}} = 20228,2 + 3992,5 = 24220,7 \text{ руб.}$$

На основании вышеприведенных расчетов делаем вывод, что основная и дополнительно заработная плата работников по созданию web приложения «Изучение английского языка по средствам флэш карточек» составит 20228,2 руб.

3.1.3 Расчёт отчислений на социальное страхование

Отчисления с заработной платы на социальное страхование во внебюджетные фонды включает следующие виды отчислений:

- фонд социального страхования — 24,9 %;
- фонд обязательного медицинского страхования — 5,1%;
- страхование от несчастных случаев или профессиональных заболеваний — 0,2%;

Всего отчисления на социальное страхование составляют 30% от суммы основной и дополнительной заработной платы. Отчисления на социальное страхование рассчитываются по формуле 4.

$$З_{\text{стр}} = ЗП_{\text{сум}} \times 30,2 / 100, \quad (4)$$

где $ЗП_{\text{сум}}$ — сумма основной и дополнительной заработной платы, руб.;

Значение $ЗП_{\text{осн}}$ возьмём из таблицы 3, а $ЗП_{\text{доп}}$ рассчитывается по формуле 3. Вычислим величину отчислений с помощью формулы 4.

$$З_{\text{стр}} = 24220,7 \times 30,2 / 100 = 7314,65 \text{ руб.}$$

Таким образом, из расчетов видно, что отчисления на социальное страхование составят 7314,65 руб.

3.2 Расчёт стоимости материалов и лицензионного обеспечения

Расчёт стоимости необходимых материалов для разработки программного продукта рассчитывается на основе норм расхода материальных ресурсов и оптовых цен на их приобретение по формуле 5.

$$C_m = N_p \times Ц, \quad (5)$$

где N_p — норма расхода материальных ресурсов в натуральных единицах;

$Ц$ — цена приобретения за единицу, руб.

Расчёт стоимости материалов представлен в таблице 4.

Таблица 4 — Расчёт стоимости материалов в рублях

Наименование материала	Норма расхода	Цена за единицу	Сумма
USB 3.0 Флеш-накопитель, шт.	1	380	380
Бумага, уп.	1	550	550
Картридж, шт.	1	2000	2000
Папка, шт.	1	160	160
Итого			3090

Общая стоимость материалов составит 3090 руб.

При создании программного продукта используется лицензионное программное обеспечение, поэтому необходимо включить стоимость лицензионных программ, используемых при разработке программного продукта, в смету затрат.

Сначала определим перечень лицензионных программ, срок их действия и норму установки в таблице 5.

Таблица 5 — Перечень программного обеспечения для выполнения работ

Наименование лицензионных программ	Норма установки, шт.	Цена, руб.
Microsoft Windows 11 Pro	1	4900
Microsoft Office 2021	1	5100
Wireframe cc	1	В свободном доступе
Visual Studio Code	1	В свободном доступе
Open Server Panel	1	В свободном доступе
Итого		10000

Стоимость лицензионного программного обеспечения для разработки составит 10000 руб.

Итого прямых расходов — $24220,7 + 7314,65 + 3090 + 10000 = 44625,35$ руб.

3.3 Расчёт накладных расходов

Накладные расходы представляют собой дополнительные к основным расходам затраты на управление, организацию и обслуживание производства.

Не связаны напрямую с основным производством товаров или предоставлением услуг, не входят в стоимость материалов и оплату труда.

Накладные расходы закладываются в себестоимость товара, издержки его производства и обращения, но не прямо, а косвенно, пропорционально стоимости материалов и сырья, сумме заработной платы и так далее.

Накладные расходы составляют 10% от расходов на эксплуатацию и рассчитываются по формуле 8.

$$H_p = (P_{\text{эксп}} \times 10) / 100, \quad (8)$$

где $P_{\text{эксп}}$ — величина расходов на эксплуатацию, руб.

$$H_p = (44625,35 \times 10) / 100 = 4462,53 \text{ руб.}$$

На основании выше рассчитанных составлена смета затрат на программный продукт. Таблица 6 и Рисунок Д.1.

Таблица 6 — Смета затрат на программный продукт

Статьи затрат	Сумма, руб.	Структура затрат, %
Основная и дополнительная заработная плата	24220,7	49,34
Отчисления на социальное страхование	7314,65	14,9
Стоимость материалов	3090	6,29
Стоимость лицензионного ПО	10000	20,37
Итого расходы на эксплуатацию	44625,35	90,91
Накладные расходы	4462,53	9,09
Итого	49087,88	100

Таким образом, общая сумма затрат на разработку web приложения по изучению английского языка по средствам флэш карточек составит 49087,88 руб.

3.4 Составление и расчёт цены реализации программного продукта

Цена является одним из самых важных экономических показателей. Её основная функция состоит в обеспечении выручки от реализации программного продукта, поэтому цена определяет прибыль и финансовую стабильность предприятия, его жизнеспособность.

Так как целью предпринимательской деятельности является получение прибыли, то цена реализации программного продукта будет представлять собой сумму затрат на его разработку и запланированную величину прибыли. Примем уровень рентабельности $P = 30\%$. Величину общих затрат на разработку программного продукта возьмём из таблицы 6, а стоимости материалов из таблицы 4.

Расчёт оптовой цены производится по формуле 9.

$$Ц_{\text{опт}} = 3 \times (1 + P), \quad (9)$$

где 3 — общие затраты на разработку программного продукта, руб.;

P — уровень рентабельности проекта, коэффициент.

Произведём расчёт оптовой цены используя формулу 9

$$Ц_{\text{опт}} = 49087,88 \times (1 + 0,30) = 63814,24 \text{ руб.}$$

Расчёт прибыли от продажи продукта производится по формуле 10:

$$\Pi = Ц_{\text{опт}} - 3, \quad (10)$$

$$\Pi = 63814,24 - 49087,88 = 14726,36 \text{ руб.}$$

Таким образом, можно сделать вывод, что данный продукт прибыльный и может быть реализован. Прибыль от продажи программного продукта составит 14726 руб. 36 коп.

Разработка веб-приложения для изучения английского языка с использованием флэш-карточек представляет собой важную задачу, поскольку английский язык является одним из наиболее широко используемых языков в мире. Приложение, созданное в рамках данной работы, обеспечивает удобный доступ к обучающим материалам, позволяет эффективно запоминать новые слова и выражения, а также повышать уровень владения английским языком.

В процессе выполнения дипломной работы были решены следующие задачи:

- сформулирована постановка задачи и определена необходимость разработки веб-приложения для изучения английского языка с использованием флэш-карточек.
- описаны технические и математические аспекты задачи с использованием соответствующих диаграмм и моделей;
- проведен обзор и анализ существующих аналогов, предоставляющих подобные функции, выявлены их преимущества и недостатки, что позволило определить основные требования к разрабатываемому веб-приложению;
- обоснованы проектные решения, связанные с программным, информационным и техническим обеспечением приложения, определены требования к программе, разработана информационная модель, представлена структура базы данных, а также спроектированы макеты пользовательского интерфейса;
- представлены алгоритмы, описывающие работу приложения, включая методы обработки и отображения флэш-карточек, реализованы меры безопасности для защиты данных пользователей;
- описаны функциональное назначение приложения, входные и выходные данные, а также особенности работы пользователя и администратора с разработанным веб-приложением;
- рассмотрены вопросы безопасности при использовании приложения на компьютере.

Результатом данной дипломной работы является веб-приложение, предоставляющее пользователю удобный и эффективный способ изучения английского языка с помощью флэш-карточек. Разработанный проект отвечает поставленным требованиям, а его функциональность и удобство использования позволят пользователям значительно повысить свой уровень владения английским языком.

1. ГОСТ 2.105–2019. Общие требования к текстовым документам. – Москва: Стандартинформ, 2021. – 33 с.
2. Дакетт, Джон HTML и CSS. Разработка и дизайн веб-сайтов Джон Дакетт. – Москва: Наука, 2014. – 480 с.
3. Евдокимов, П. В. С# на примерах / П. В. Евдокимов. – Санкт-Петербург: Наука и техника, 2016. – 304 с.
4. Дронов, В.П. JavaScript в Web-дизайне / В.П. Дронов. – М.: Санкт-Петербург: БХВ, 2014. – 880 с.
5. Флёнов, М.Е. Библия С#: учебное пособие / М.Е. Флёнов. – 3-е изд., исп. и доп. – Санкт-Петербург: БХВ-Петербург, 2016. – 544 с.
6. Дронов, В.П. PHP, MySQL и Dreamweaver. Разработка интерактивных Web-сайтов / В.П. Дронов. – Москва: БХВ-Петербург, 2014. – 480 с.
7. Дронов, В.П. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов / В.П. Дронов. – Москва: БХВ–Петербург, 2015. – 416 с.
8. Шкрыль, А.А. PHP – это просто. Программируем для Web-сайта / А.А. Шкрыль. – Москва: БХВ–Петербург, 2015. – 368 с.
9. Кузнецов М.В. PHP. Практика создания Web-сайтов / Кузнецов, М.В. и. – Москва: БХВ-Петербург, 2016. – 895 с.
10. Гончаров, А.А. Web-дизайн / А.А Гончаров. – Москва: ИЛ, 2016. – 320 с.
11. База ГОСТов. ГОСТ Р 2.105-2019 Единая система конструкторской документации. Общие требования к текстовым документам.

Приложение А
(обязательное)



Рисунок А.1 – Диаграмма декомпозиции IDEF0

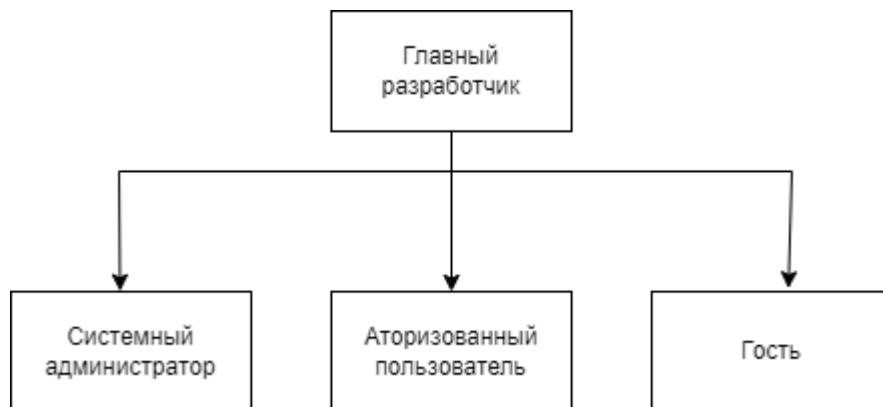


Рисунок А.2 – Организационная диаграмма

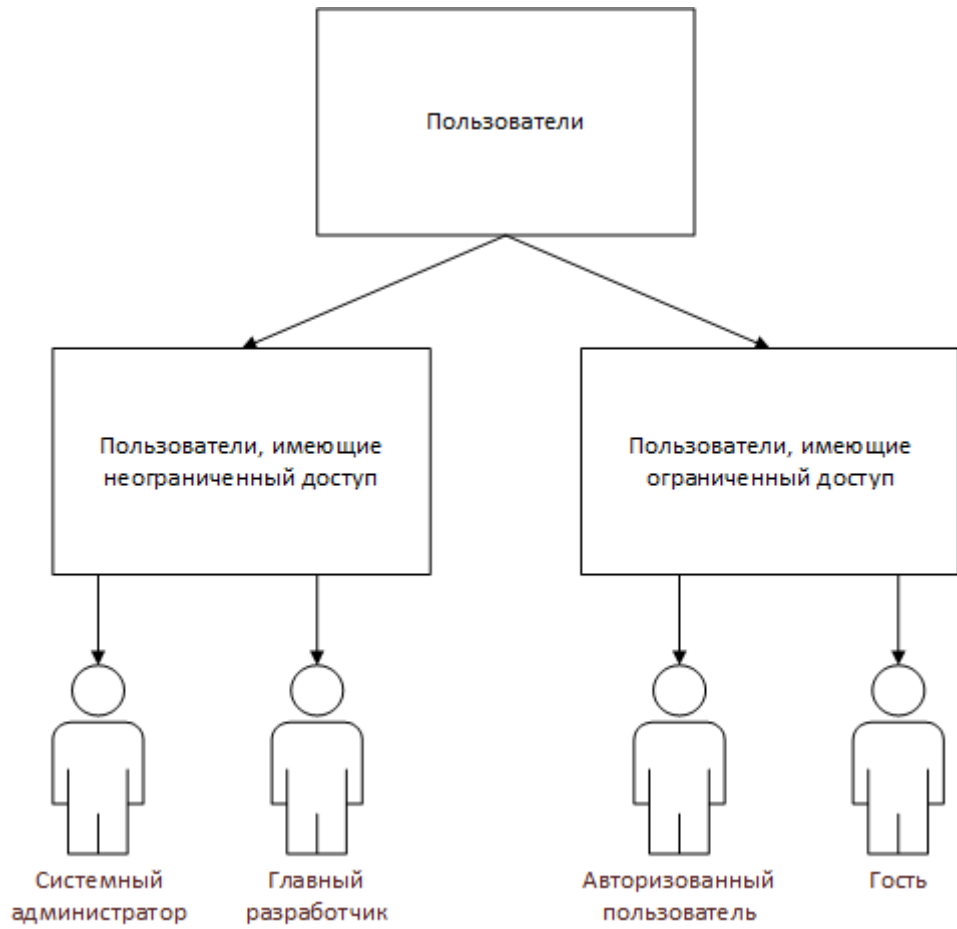


Рисунок А.3 – Группы пользователей системы



Рисунок А.4 – Диаграмма бизнес-прецедентов

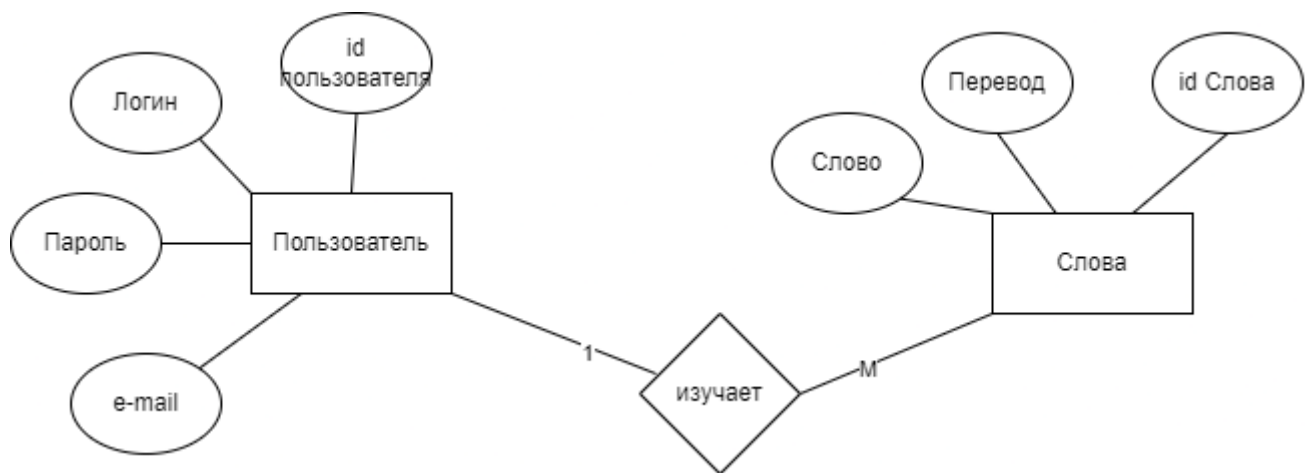


Рисунок А.5 – Entity-Relationship модель базы данных

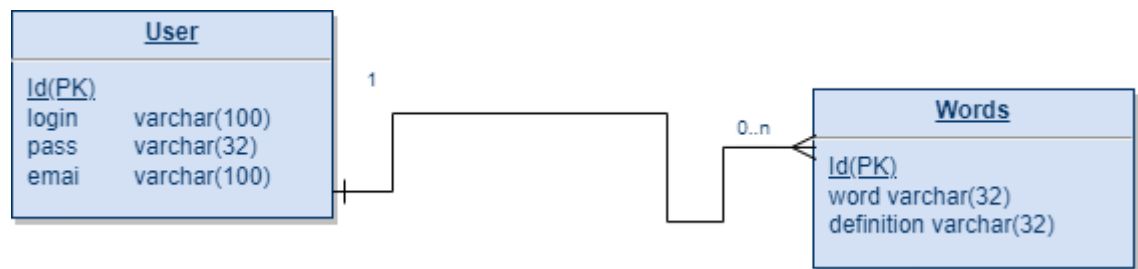


Рисунок А.6 – Реляционная модель данных

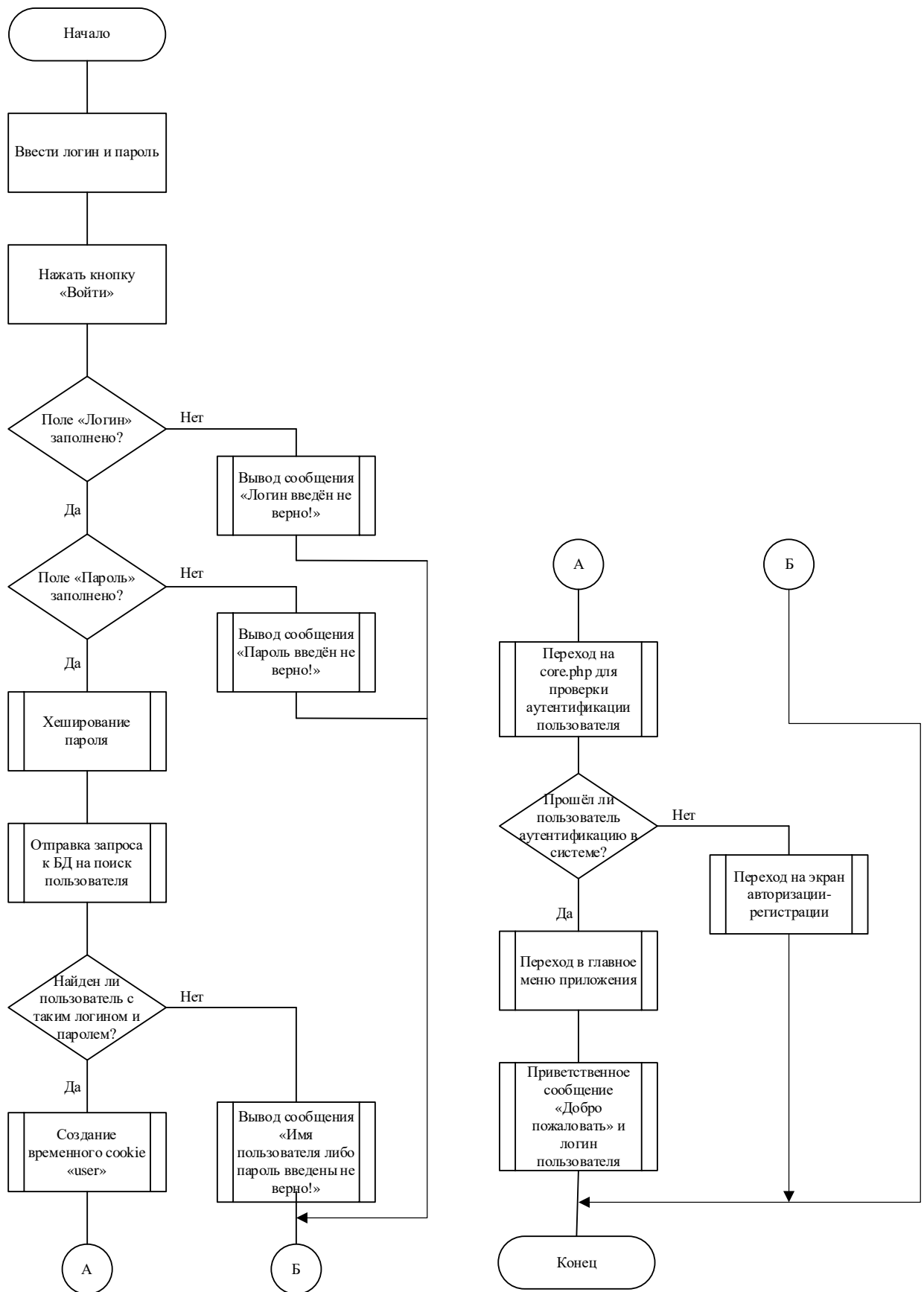


Рисунок А.7 – Алгоритм метода auth.php

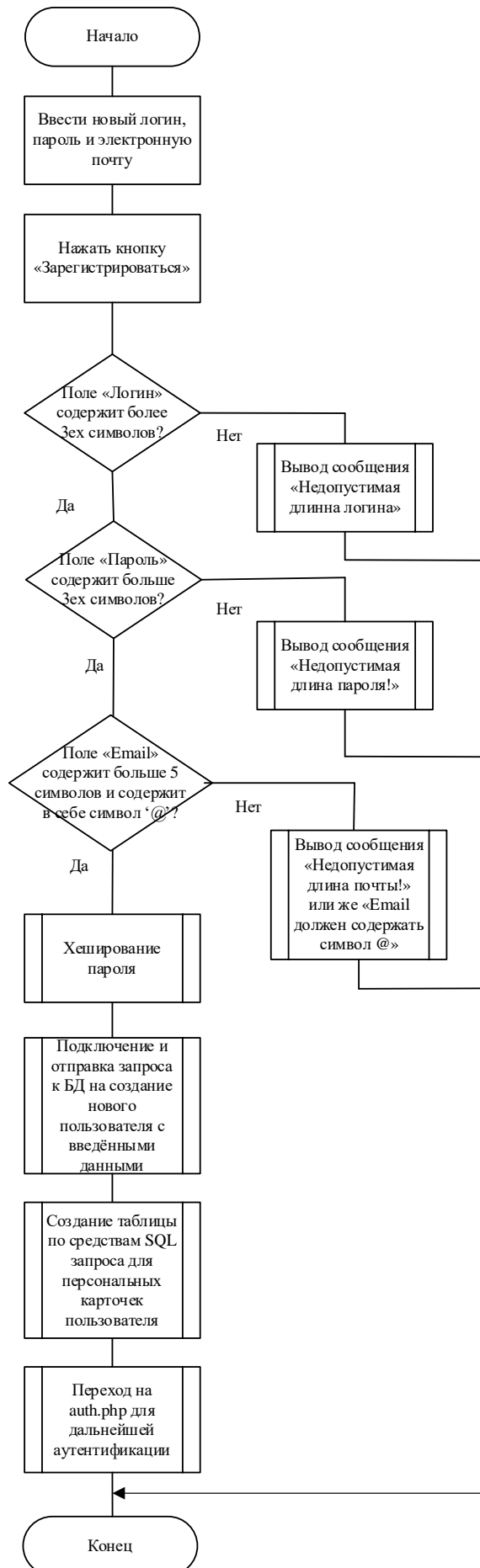
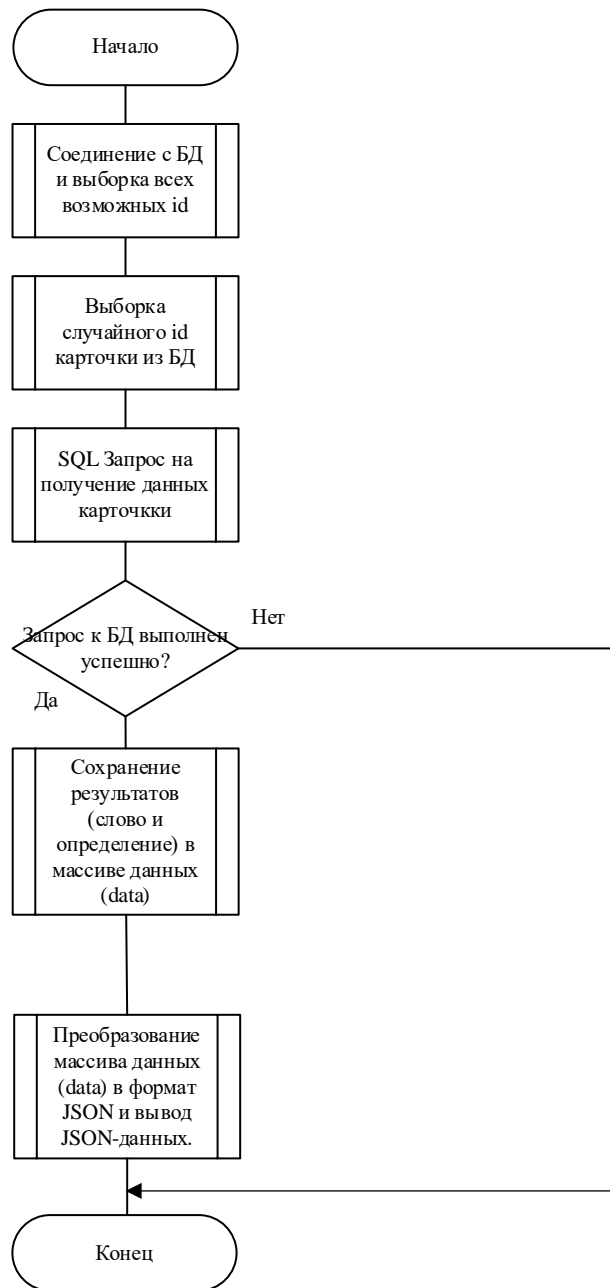


Рисунок А.7 – Алгоритм метода reg.php

Рисунок А.8 – Алгоритм метода `updateWords.php`

Приложение Б

(обязательное)

Б.1 – Скрипт создания таблицы с данными пользователей

```
CREATE TABLE `user` (  
  `id` int(11) UNSIGNED NOT NULL,  
  `login` varchar(100) NOT NULL,  
  `pass` varchar(32) NOT NULL,  
  `email` varchar(100) NOT NULL  
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

Б.2 – Скрипт создания таблиц со словами

```
CREATE TABLE `words` (  
  `id` int(11) UNSIGNED NOT NULL,  
  `word` varchar(32) CHARACTER SET utf8 NOT NULL,  
  `definition` varchar(32) CHARACTER SET utf8 NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Приложение В

(обязательное)

В.1 – Регистрация, метод reg.php

```
<?php
$login = filter_var(trim($_POST['loginR']), FILTER_SANITIZE_STRING);
$pass = filter_var(trim($_POST['passR']), FILTER_SANITIZE_STRING);
$email = filter_var(trim($_POST['email']), FILTER_SANITIZE_STRING);

if (mb_strlen($login) < 3 || mb_strlen($login) > 90) {
    echo "<script>
    window.location = '/core.php';
    alert('Недопустимая длинна логина');
    </script>";
    exit();
} else if (mb_strlen($pass) < 3 || mb_strlen($pass) > 50) {
    echo "<script>
    window.location = '/core.php';
    alert('Недопустимая длина пароля');
    </script>";
    exit();
} else if (mb_strlen($email) < 5 || mb_strlen($email) > 100) {
    echo "<script>
    window.location = '/core.php';
    alert('Недопустимая длина почты');
    </script>";
    exit();
}
else if (strpos($email, '@') === false) {
    echo "<script>
    window.location = '/core.php';
    alert('Email должен содержать символ @');
    </script>";
    exit();
}

$pass = md5($pass . "regregj343");

$mysql = new mysqli('172.20.8.5:3306', 'st9903_11', 'pwd9903_11', 'st9903_11');

// Проверяем наличие пользователя с таким логином в базе данных
$existingUser = $mysql->query("SELECT * FROM `user` WHERE `login` = '$login'");

if ($existingUser->num_rows > 0) {
    echo "<script>
    window.location = '/core.php';
    alert('Пользователь с таким логином уже существует');
    </script>";
    exit();
}

// Регистрация нового пользователя
$mysql->query("INSERT INTO `user` (`login`, `pass`, `email`) VALUES ('$login',
'$pass', '$email')");
$mysql->query("CREATE TABLE `words_$login` (
```

```

        `id` int(11) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
        `word` varchar(32) CHARACTER SET utf8 NOT NULL,
        `definition` varchar(32) CHARACTER SET utf8 NOT NULL
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
");

$mysql->close();

header('Location: /');
?>

```

В.2 – Авторизация, метод auth.php

```

<?php
    $login = filter_var(trim($_POST['loginA']),
        FILTER_SANITIZE_STRING);
    $pass = filter_var(trim($_POST['passA']),
        FILTER_SANITIZE_STRING);

    $pass = md5($pass."regregj343");

    $data = 'Hello, world!';
    echo "<script>console.log('$data');</script>";
    echo "<script>console.log('$login');</script>";
    echo "<script>console.log('$pass');</script>";

    $mysql = new mysqli('172.20.8.5:3306', 'st9903_11', 'pwd9903_11', 'st9903_11');

    $result = $mysql->query("SELECT * FROM `user` WHERE `login` =
    '$login' AND `pass` = '$pass'");
    $user = $result->fetch_assoc();
    if($user == 0) {
        echo "<script>alert('Имя пользователя либо пароль введены неверно');
        window.location = '/core.php';
        </script>";
    }
    else {
        setcookie('user', $user['login'], time() + 3600, "/");

        $mysql->close();

        // header('Location: /core.php');

        echo "<script>
            if (document.referrer.indexOf('ru/index.php') > -1) {
                window.location.href = '/ru/core.php';
            }
            else {
                window.location.href = '/core.php';
            }

            </script>";

    }

?>

```

В.3 – Выход из аккаунта, метод exit.php

```
<?php
    setcookie('user', $user['login'], time() - 3600, "/");
    header('Location: /core.php');
?>
```

В.4 – Добавление слова, метод addWord.php

```
<?php
    $word = filter_var(trim($_POST['word']),
        FILTER_SANITIZE_STRING);
    $definition = filter_var(trim($_POST['definition']),
        FILTER_SANITIZE_STRING);

    if (empty($word) || empty($definition)) {
        echo "<script>
            window.location = '/app.php';
            alert('Вы не заполнили одно из значений');
        </script>";
        exit();
    }

    $mysql = new mysqli('172.20.8.5:3306', 'st9903_11', 'pwd9903_11', 'st9903_11');
    $user = $_COOKIE['user'];

    $mysql->query("INSERT INTO `words_$user` (`word`,`definition`)
    VALUES('$word','$definition')");
    //Тест на ошибки к БД запросы
    // if ($mysql->errno) {
    //     echo "Ошибка базы данных: " . $mysql->error;
    // }

    echo "<script>
        if (document.referrer.indexOf('/ru/app.php') > -1) {
            window.location = '/ru/app.php';
        }
        else if (document.referrer.indexOf('/ru/words.php') > -1) {
            window.location = '/ru/words.php';
        }
        else if (document.referrer.indexOf('/words.php') > -1) {
            window.location = '/words.php';
        }
        else {
            window.location.href = '/app.php';
        }

        </script>";
?>
```


B.5 – Удаление слова, метод wordDel.php

```
<?php
$word = filter_var(trim($_POST['word']), FILTER_SANITIZE_STRING);
$user = $_COOKIE['user'];

mysql = new mysqli('172.20.8.5:3306', 'st9903_11', 'pwd9903_11', 'st9903_11');

mysql->query("DELETE FROM `words_$user` WHERE `word` = '$word'");

echo "<script>
if (document.referrer.indexOf('/ru/app.php') > -1) {
    window.location.href = '/ru/app.php';
}
else if (document.referrer.indexOf('/ru/words.php') > -1) {
    window.location = '/ru/words.php';
}
else if (document.referrer.indexOf('/words.php') > -1) {
    window.location = '/words.php';
}
else {
    window.location.href = '/app.php';
}
</script>";
?>
```

B.6 – Редактирование карточки, метод editWord.php

```
<?php
$seditValue = filter_var(trim($_POST['edit']), FILTER_SANITIZE_STRING);
$change = filter_var(trim($_POST['change']), FILTER_SANITIZE_STRING);
$defChange = filter_var(trim($_POST['defChange']), FILTER_SANITIZE_STRING);
$user = $_COOKIE['user'];

mysql = new mysqli('172.20.8.5:3306',
'st9903_11', 'pwd9903_11', 'st9903_11');

mysql->query("UPDATE `words_$user` SET `word` = '$change', `definition` =
'$defChange' WHERE `word` = '$seditValue'");

echo "<script>
if (document.referrer.indexOf('/ru/app.php') > -1) {
    window.location.href = '/ru/app.php';
}
else if (document.referrer.indexOf('/ru/words.php') > -1) {
    window.location = '/ru/words.php';
}
else if (document.referrer.indexOf('/words.php') > -1) {
    window.location = '/words.php';
}
else {
    window.location.href = '/app.php';
}
</script>";
?>
```

B.7 – Обновление флэш карточки, метод updateWords.php

```
<?php
    $mysql = new mysqli('172.20.8.5:3306', 'st9903_11', 'pwd9903_11',
'st9903_11');
    $user = $_COOKIE['user'];

    $sqlIDs = "SELECT id FROM words_$user";
    $idResult = $mysql->query($sqlIDs);
    $idArray = array();

    while ($row = $idResult->fetch_assoc()) {
        $idArray[] = $row['id'];
    }

    $idResult->free();

    $randomIndex = array_rand($idArray);
    $randomId = $idArray[$randomIndex];

    $sql = "SELECT word, definition FROM words_$user WHERE id = '$randomId'";

    if ($result = $mysql->query($sql)) {
        $rowsCount = $result->num_rows;
        $data = array();

        foreach ($result as $row) {
            $data[] = array(
                'word' => $row['word'],
                'definition' => $row['definition']
            );
        }

        $result->free();
    }

    echo json_encode($data);
?>
```

B.8 – Просмотр словаря, words.php

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <meta http-equiv="X-UA-Compatible" content="ie=edge">
        <title>Dictionary</title>
        <link rel="icon" href="img/flash-cards.png" type="image/x-icon">
        <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css">
        <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css'
rel='stylesheet'>
        <link href="/css/words.css" rel="stylesheet" type="text/css">
    </head>

    <body>
        <header class="header">
            <nav class="navbar">
```

```

        <a href="/core.php/">Home</a>
        <a href="/info.php#about">About</a>
        <a href="/info.php#services">Services</a>
        <a href="/info.php#contact">Contact</a>
    </nav>

    <div class="langSel">
        <select id="language-selector">
            <option value="english">English</option>
            <option value="russian">Русский</option>
        </select>
    </div>
</header>

<?php
    $mysql = new mysqli('172.20.8.5:3306',
'st9903_11','pwd9903_11','st9903_11');
    $user = $_COOKIE['user'];

    if($mysql->connect_error){
        die("Ошибка: " . $mysql->connect_error);
    }
    $sql = "SELECT * FROM `words_$user`";
    if($result = $mysql->query($sql)){
        $rowCount = $result->num_rows;
        echo "<div class= table-container>";
        echo "<h1>A list of your words</h1>";
        echo "<p>Words in your dictionary: $rowCount</p>";
        echo "<hr/>";
        echo "<table><tr><th>Term</th><th>Definition</th></tr>";
        foreach($result as $row){
            echo "<tr>";
                echo "<td>" . $row["word"] . "</td>";
                echo "<td>" . $row["definition"] . "</td>";
            echo "</tr>";
        }
        echo "</table>";
        $result->free();
    } else{
        echo "Ошибка: " . $mysql->error;
    }
    ?>

    <div class="but">
        <button class="btn btn-danger" id="delWord" role="button"> Delete <i
class='bx bx-trash'></i></button>
        <button class="btn btn-warning" id="edWord" role="button"> Edit <i
class='bx bxs-edit'></i></button>
        <button class="btn btn-success" id="addWord" role="button"> Add <i
class='bx bx-plus-circle'></i></button>
    </div>

    <div class="modal" id="delWindow">
        <div class="modal-content">
            <button class="closeModal-btn" id="close-btnDel"><i class='bx bx-x'
style='color:#ff0000' ></i></button>
            <h2>Word deletion</h2>
            <form action="/actions/wordDel.php" method="post">
                <div class="termD">
                    <h3>Are you sure that you wanna delete <span id="output"></span>
word?</h3>
                    <input type="hidden" name="word" id="wordInput" value="">
                </div>

```

```

        <button class="btn btn-outline-danger"
type="submit">Delete</button>
        <button id="cancel-btnDel" type="button" class="btn btn-outline-
primary">Cancel</button>
    </form>
</div>
</div>

<div class="modal" id="edWindow">
    <div class="modal-content">
        <button class="closeModal-btn" id="close-btnEd"><i class='bx bx-x'
style='color:#ff0000' ></i></button>
        <h2>Editing the word</h2>
        <form action="/actions/editWord.php" method="post">
            <div class="termEd">
                <h3>Word: <span id="output2"></span></h3>
                <input type="hidden" name="edit" id="edInput" value="">
            </div>
                <input type="text" class="form-control" name="change"
id="change" placeholder="Enter the new term"><br>
                <input type="text" class="form-control" name="defChange"
id="defChange" placeholder="Enter the new definition"><br>
                <button class="btn btn-outline-warning"
type="submit">Edit</button>
            </form>
        </div>
    </div>

    <div class="modal" id="addWindow">
        <div class="modal-content">
            <button class="closeModal-btn" id="close-btnAdd"><i class='bx bx-x'
style='color:#ff0000' ></i></button>
            <h2>Adding the word</h2>
            <form action="actions/addWord.php" method="post">
                <input type="text" class="form-control" name="word" id="word"
placeholder="Type the word in English"><br>
                <input type="text" class="form-control" name="definition"
id="definition" placeholder="Enter translation"><br>
                <button class="btn btn-outline-success" type="submit">Add
word</button>
            </form>
        </div>
    </div>

</div>

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script src="/js/words.js"></script>
</body>
</html>

```

В.9 – Меню выбора набора карточек, selector.php

```

<!DOCTYPE html>
<html lang="en" dir="ltr">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <title>Flash Card App</title>

```

```

<link rel="icon" href="/img/flash-cards.png" type="image/x-icon">
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css">
<link rel="stylesheet" type="text/css" href="/css/selector.css">
</head>
<body>

    <header class="header">
        <nav class="navbar">
            <a href="/core.php">Home</a>
            <a href="/info.php#about">About</a>
            <a href="/info.php#services">Services</a>
            <a href="/info.php#contact">Contact</a>
        </nav>

        <div class="langSel">
            <select id="language-selector">
                <option value="english">English</option>
                <option value="russian">Русский</option>
            </select>
        </div>
    </header>

    <h1>Choose a set of cards to study</h1>
    <div class="container">
        <div class="box" id="card1">
            <span></span>
            <div class="content">
                <h2>The most common words</h2>
                <p>Here you can learn and repeat the most common words in
English every day!</p>
            </div>
        </div>

        <div class="box" id="card2">
            <span></span>
            <div class="content">
                <h2>Slang Cards</h2>
                <p>Learn slang phrases in English with this set of
cards!</p>
            </div>
        </div>

        <div class="box" id="card3">
            <span></span>
            <div class="content">
                <h2>Personal cards</h2>
                <p>Create your own, completely unique set of cards. </p>
            </div>
        </div>
    </div>

    <script src="/js/selector.js"></script>
</body>
</html>

```

В.10 – Приложение с флэш карточками, app.php

```

<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Flash Card App</title>
  <link rel="icon" href="img/flash-cards.png" type="image/x-icon">
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css">
  <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css'
rel='stylesheet'>
  <link href="css/app.css" rel="stylesheet" type="text/css">
</head>

<body>

  <header class="header">
    <nav class="navbar">
      <a href="/core.php/">Home</a>
      <a href="/info.php#about">About</a>
      <a href="/info.php#services">Services</a>
      <a href="/info.php#contact">Contact</a>
    </nav>

    <div class="langSel">
      <select id="language-selector">
        <option value="english">English</option>
        <option value="russian">Русский</option>
      </select>
    </div>
  </header>

  <?php
    $mysql = new mysqli('172.20.8.5:3306',
'st9903_11','pwd9903_11','st9903_11');

    $sqlc = "SELECT word, definition FROM words";
    $col = $mysql->query($sqlc);
    $rowCount2 = $col->num_rows;
    $min = rand(1,$rowCount2);

    $sql = "SELECT word, definition FROM words WHERE id = '$min'";

    if($result = $mysql->query($sql)){
      $rowCount = $result->num_rows;

      foreach($result as $rows2){
      }

      $result->free();
      $col->free();
    }
  ?>

  <div class="containerG">
    <div class="leftM">

```

```

        <h4> <i class='bx bxs-copy'></i> Number of cards:  <?=$rowsCount2?>
    </h4>
    </div>
    <div class = "container">
    <div class="card">
        <div class="front">
            <div class="term">
                <h3> <?php echo $rows2["word"]; ?></h3>
            </div>

            <p> <i class='bx bxs-hand-up'></i> Show definition</p>
        </div>

        <div class="back">
            <div class="definition">
                <h3> <?php echo $rows2["definition"]; ?></h3>
                <div class="termBack">
                    <p> <?php echo $rows2["word"]; ?></p>
                </div>
            </div>
        </div>
    </div>

    <div class="but">
        <a href="#" class="btn btn-outline-light" onclick="updateWords()"
        role="button"> Next word </a>
        <button class="btn btn-outline-light" id="delWord" role="button"> <i
        class='bx bx-trash'></i></button>
        <button class="btn btn-outline-light" id="addWord" role="button"> <i
        class='bx bx-plus-circle'></i></button>
    </div>
    </div>
    <div class="item">

    </div>
</div>

<div class="modal" id="delWindow">
    <div class="modal-content">
        <button class="closeModal-btn" id="close-btnDel"><i class='bx bx-x'
        style='color:#ff0000' ></i></button>
        <h2>Удаление слова</h2>
        <form action="/actions/wordDel.php" method="post">
            <input type="text" class="form-control" name="word" id="word"
            placeholder="Введите id слова для удаления"><br>
            <button class="btn btn-outline-danger" type="submit">Удалить
            слово</button>
        </form>
    </div>
</div>

<div class="modal" id="addWindow">
    <div class="modal-content">
        <button class="closeModal-btn" id="close-btnAdd"><i class='bx bx-x'
        style='color:#ff0000' ></i></button>
        <h2>Добавление слова</h2>
        <form action="actions/addWord.php" method="post">
            <input type="text" class="form-control" name="word" id="word"
            placeholder="Введите слово на английском"><br>
            <input type="text" class="form-control" name="definition"
            id="definition" placeholder="Введите перевод"><br>
            <button class="btn btn-outline-success" type="submit">Добавить
            слово</button>
        </form>
    </div>
</div>

```

```

        </div>
    </div>

    <script src="js/app.js"></script>
</body>
</html>

```

B.11 – Форма регистрации-авторизации, index.php

```

<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Flash Card App</title>
    <link rel="icon" href="/img/flash-cards.png" type="image/x-icon">
    <link rel="stylesheet" type="text/css" href="css/styles.css">
    <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css'
rel='stylesheet'>
  </head>

  <body>

    <header class="header">
      <nav class="navbar">
        <a href="/core.php">Home</a>
        <a href="/info.php#about">About</a>
        <a href="/info.php#services">Services</a>
        <a href="/info.php#contact">Contact</a>

      </nav>

      <div class="langSel">
        <select id="language-selector">
          <option value="english">English</option>
          <option value="russian">Русский</option>
        </select>
      </div>
    </header>

    <div class="container">
      <div class="content">
        <h2 class="logo">  Flash Card App </h2>

        <div class="text-sci">
          <h2>Welcome!<br><span>To our new Website for learning</span></h2>

          <p> Here, you can start your English learning journey by flash cards.
We wish you success!</p>

          <div class="social-icons">
            <a href=""><i class='bx bxl-instagram'></i></a>
            <a href=""><i class='bx bxl-twitter' ></i></a>
            <a href=""><i class='bx bxl-vk' ></i></a>
            <a href=""><i class='bx bxl-facebook'></i></a>
          </div>
        </div>

        <div class="logreg-box">
          <div class="form-box login">
            <form action="actions/auth.php" method="post">
              <h2>Sign in</h2>

```



```

<div class="input-box">
  <span class="icon"><i class='bx bx-envelope'></i></span>
  <input type="text" required name="loginA" >
  <label>Login</label>
</div>

<div class="input-box">
  <span class="icon"><i class='bx bxs-lock-alt' ></i></span>
  <input type="password" required name="passA" >
  <label>Password</label>
</div>

<div class="remember-forgot">
  <label><input type="checkbox"> Remember me</label>
  <a href="#">Forgot Password?</a>
</div>

<button type="submit" class="btn">Sign in</button>

<div class="login-register">
  <p>Don't have an account? <a href="#" class="register-link">Sign
up</a></p>
</div>
</form>
</div>

<div class="form-box register">
  <form action="actions/reg.php" method="post">
    <h2>Sign Up</h2>

    <div class="input-box">
      <span class="icon"><i class='bx bx-user'></i></span>
      <input type="text" required name="loginR">
      <label>Login</label>
    </div>

    <div class="input-box">
      <span class="icon"><i class='bx bx-envelope' ></i></span>
      <input type="text" required name="email">
      <label>Email</label>
    </div>

    <div class="input-box">
      <span class="icon"><i class='bx bxs-lock-alt' ></i></span>
      <input type="password" required name="passR">
      <label>Password</label>
    </div>

    <div class="remember-forgot">
      <label><input type="checkbox"> I agree to the terms &
conditions</label>
    </div>

    <button type="submit" class="btn">Sign Up</button>

    <div class="login-register">
      <p>Already have an account? <a href="#" class="login-link">Sign
in</a></p>
    </div>
  </form>
</div>
</div>
</div>

```

```

    <script src="js/script.js"></script>
  </body>
</html>

```

B.12 – Форма главного меню, core.php

```

<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Flash Card App</title>
    <link rel="icon" href="/img/flash-cards.png" type="image/x-icon">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css">
    <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css'
rel='stylesheet'>
    <link rel="stylesheet" type="text/css" href="/css/core.css">
  </head>

  <body>

    <header class="header">
      <nav class="navbar">
        <a href="#">Home</a>
        <a href="/info.php#about">About</a>
        <a href="/info.php#services">Services</a>
        <a href="/info.php#contact">Contact</a>
      </nav>

      <div class="langSel">
        <select id="language-selector">
          <option value="english">English</option>
          <option value="russian">Русский</option>
        </select>
      </div>
    </header>

    <?php
      if($_COOKIE['user'] == ''):
        header('Location: /index.php');
      ?>

      <?php else: ?>
    <div class = "table-container">
      <div class="content">
        <div class="hello-user">
          <h4>Hello <?=$_COOKIE['user']?></h4>
          <a href="/actions/exit.php" class="btn btn-danger"
role="button">Exit <i class='bx bx-door-open'></i></a>
        </div>
        <hr/>
        <h1>Welcome to the Flash Cards app for learning English</h1><br>
        <h3>Where would you like to start?</h3><br>

        <div class="but-func">
          <a href="/words.php" class="btn btn-warning" role="button"> <i
class='bx bx-book-bookmark'></i> My Dictionary</a>
          <a href="/selector.php" class="btn btn-success" role="button"> <i
class='bx bxs-carousel'></i> Flash Cards</a>
        </div>

```

```

    </div>
    <?php endif; ?>
  </div>
  <script src="/js/core.js"></script>
</body>
</html>

```

V.13 – Форма информационной страницы проекта, info.php

```

<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Flash Card App</title>
    <link rel="icon" href="img/flash-cards.png" type="image/x-icon">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css">
    <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css'
rel='stylesheet'>
    <link rel="stylesheet" type="text/css" href="/css/info.css">
  </head>

  <body>

    <header class="scroll-header">
      <nav class="navbar">
        <a href="/core.php">Home</a>
        <a href="/info.php#about">About</a>
        <a href="/info.php#services">Services</a>
        <a href="/info.php#contact">Contact</a>
      </nav>

      <div class="navFunc">
        <a href="/ru/core.php" id="getStarted" class="btn btn-success"
role="button"> <i class='bx bxs-carousel'></i> Get started!</a>
        <div class="langSel">
          <select id="language-selector">
            <option value="english">English</option>
            <option value="russian">Русский</option>
          </select>
        </div>
      </div>
    </header>

    <div class = "table-container" id="about">
      <div class="content">
        <h1 class = "title"><i class='bx bx-info-circle bx-tada'
style='color:#d86069' ></i> About Us:</h1>
        <hr/>
        <h2>Welcome to the Flash Card App, our website is a web application
where the user can
        learn English through flash cards.
        </h2><br>

        <h2> </h2>

        <h1 class="title" id="services"><i class='bx bxs-basket'
style='color:#3d80bd' ></i> Services provided</h1>
        <hr/>

```

```

    <h2>Our application provides a free, convenient application with the
    ability to work with the database, create, delete and
    edit your flash cards with an easy-to-use, minimalist design.
  </h2><br>

```

```

  <h2> </h2></br>

```

```

    <div class="contact" id="contact">
      <h1 class="title"><i class='bx bxs-phone-call bx-tada'
style='color:#6fd7a3' ></i> Contact Us</h1>
      <hr/>
      <h2>You can contact us in the following ways: <h2>
      <h2>Contact phone number: 8 (964)-311-01-47 </h2>
      <div class="social-icons">
        <a
href="https://instagram.com/untily0ung?igshid=MzRlODBiNWFlZA=="><i class='bx
bxl-instagram' style='color:#d9174a' ></i></a>
        <a href=""><i class='bx bxl-twitter' ></i></a>
        <a href="https://vk.com/oo00o00o"><i class='bx bxl-vk' ></i></a>
      </div>

      </h2></br>
    </div>
  </div>
  <script src="/js/info.js"></script>
</body>
</html>

```

Приложение Г (обязательное)

The screenshot shows the registration page of the 'Flash Card App'. The page has a blue header with navigation links: 'Главная', 'О нас', 'Услуги', 'Связь с нами', and a language selector 'Русский'. The main content area features a background image of a British flag and a pen. On the left, there is a welcome message: 'Добро пожаловать! На наш новый веб-сайт для обучения' and a short paragraph about the app. On the right, the 'Регистрация' (Registration) form is displayed, including fields for 'Логин' (Login), 'Email', and 'Пароль' (Password), a checkbox for 'Я согласен с правилами и условиями', a red 'Зарегистрироваться' button, and a link 'У вас уже есть аккаунт? Войти'.

Рисунок Г.1 – Форма регистрации

The screenshot shows the login page of the 'Flash Card App'. The page layout is identical to the registration page, but the right-hand form is titled 'Войти' (Login). It includes fields for 'Логин' (Login) and 'Пароль' (Password), a checkbox for 'Запомнить меня' (Remember me), a link 'Забыли пароль?' (Forgot password?), a red 'Войти' button, and a link 'У вас нету аккаунта? Зарегистрироваться'.

Рисунок Г.2 – Форма авторизации

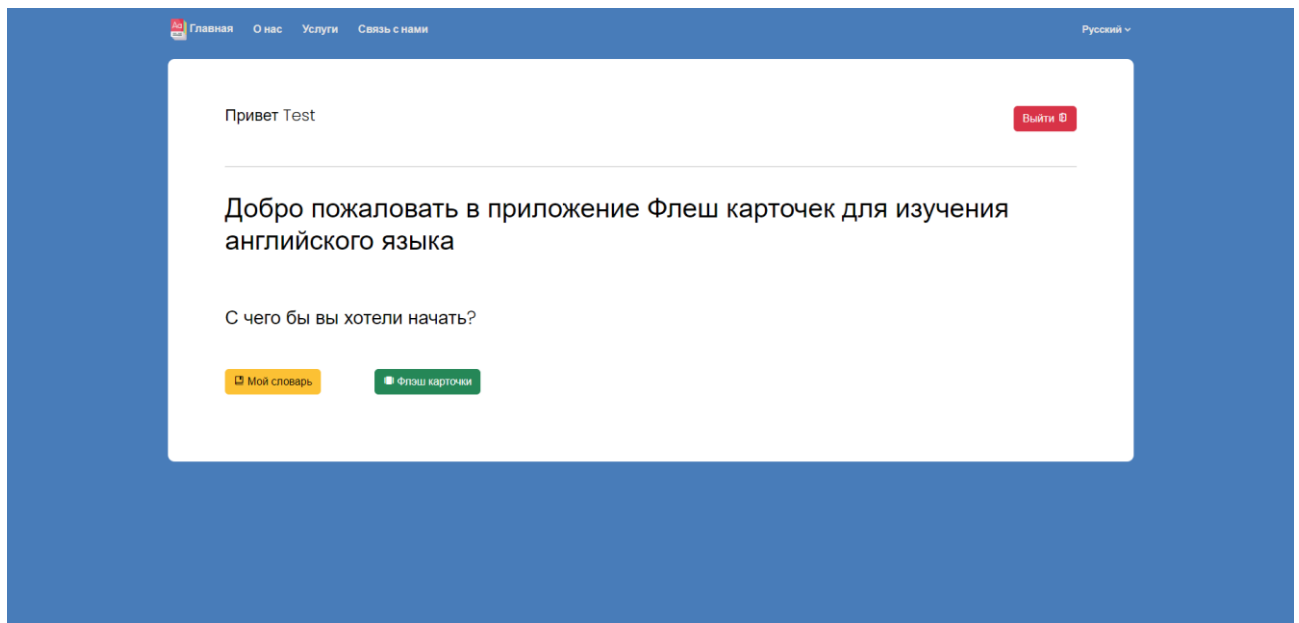


Рисунок Г.3 – Форма главного меню пользователя

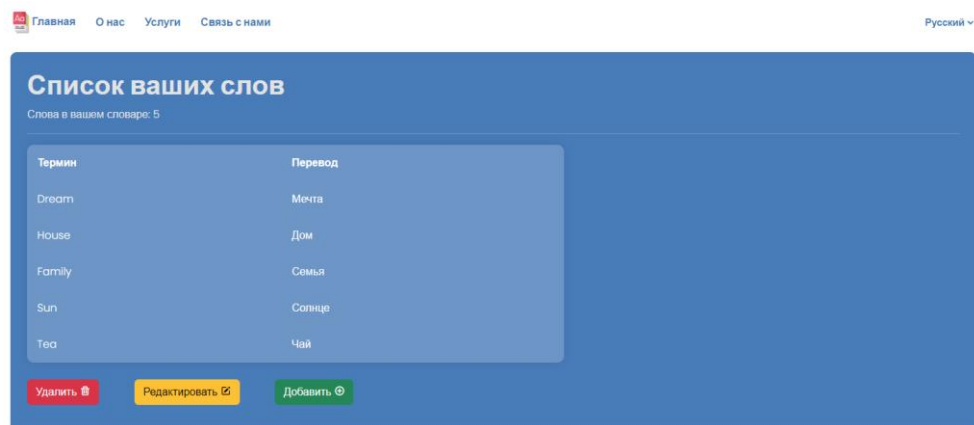


Рисунок Г.4 – Форма просмотра и редактирования словаря

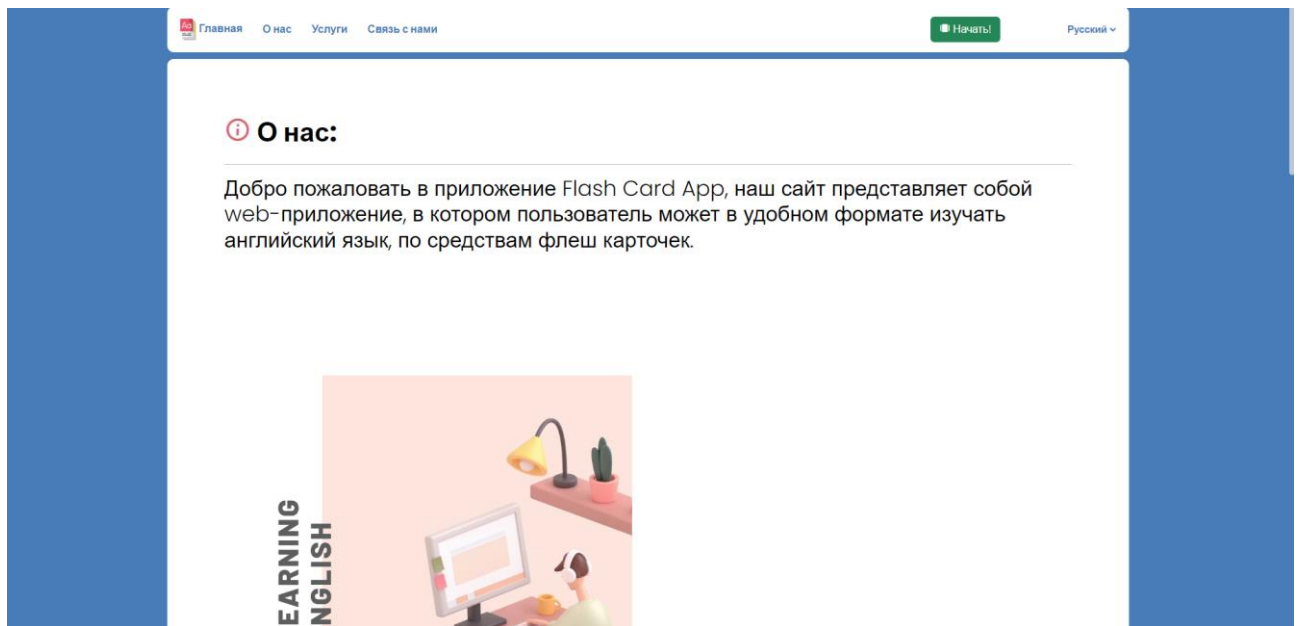


Рисунок Г.5 – Форма просмотра информации о проекте

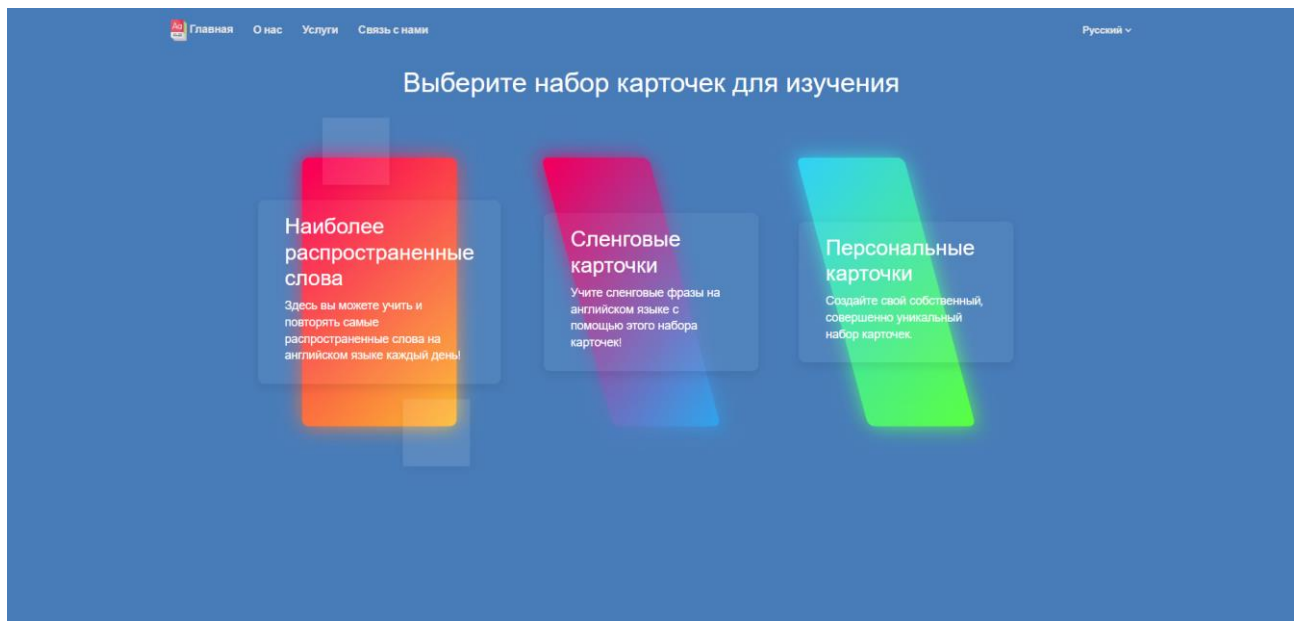


Рисунок Г.6 – Форма меню выбора набора карточек

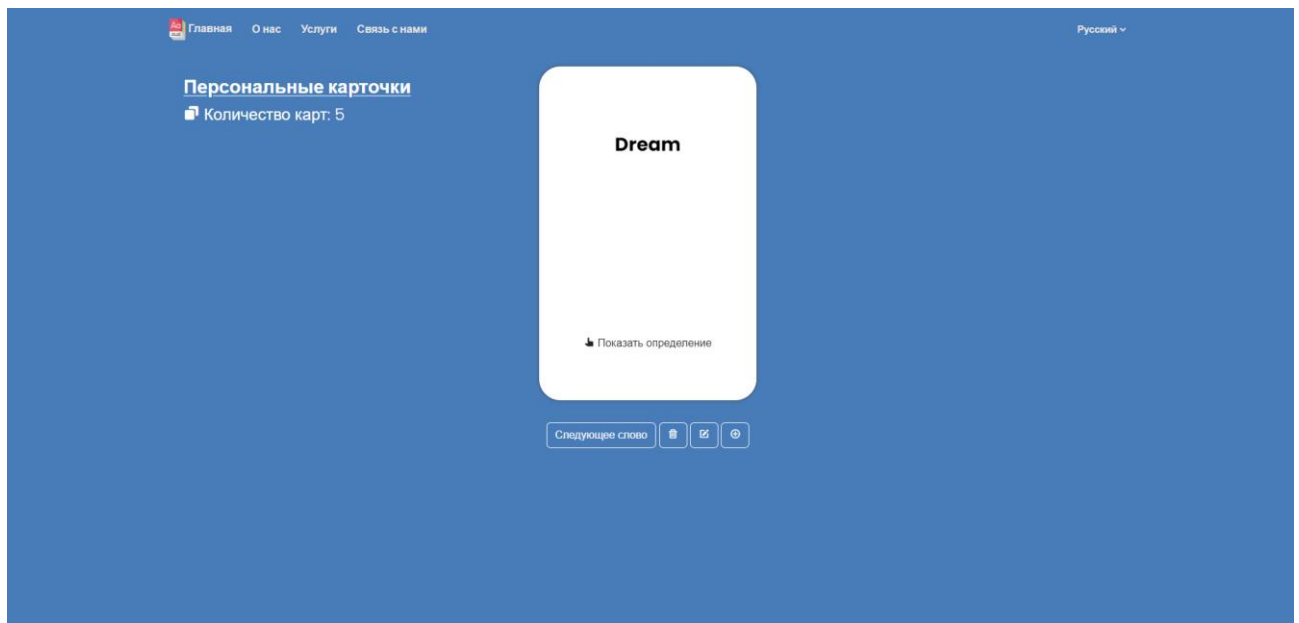


Рисунок Г.7 – Форма приложения флэш-карточек

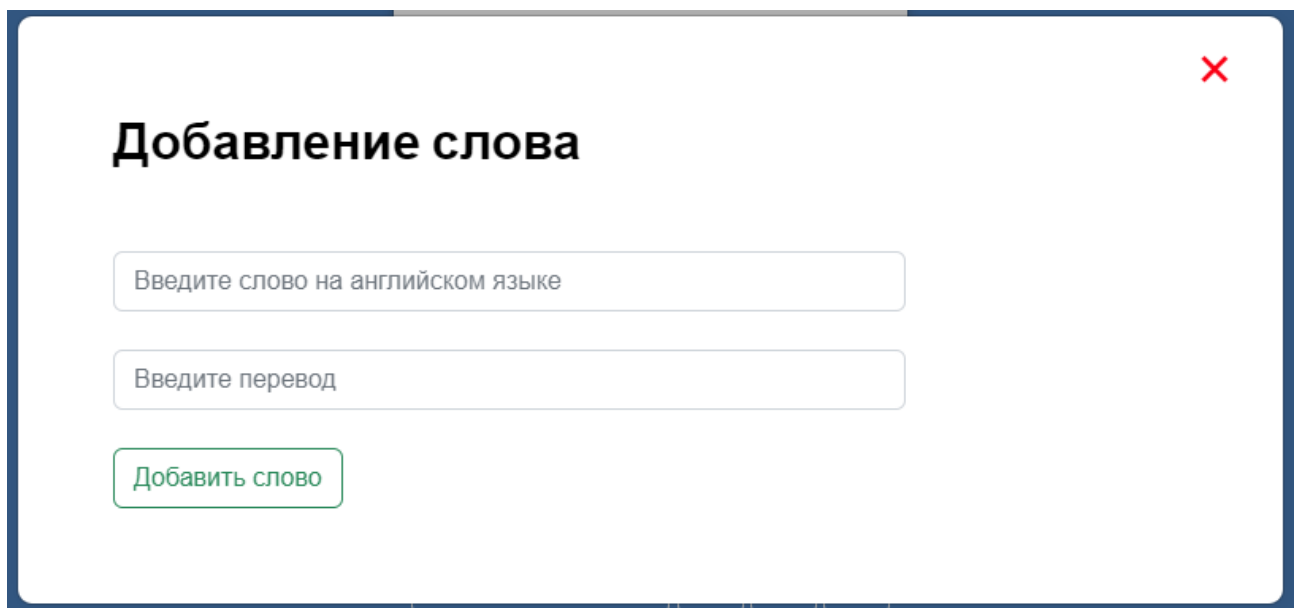


Рисунок Г.8 – Форма модального окна добавления слова

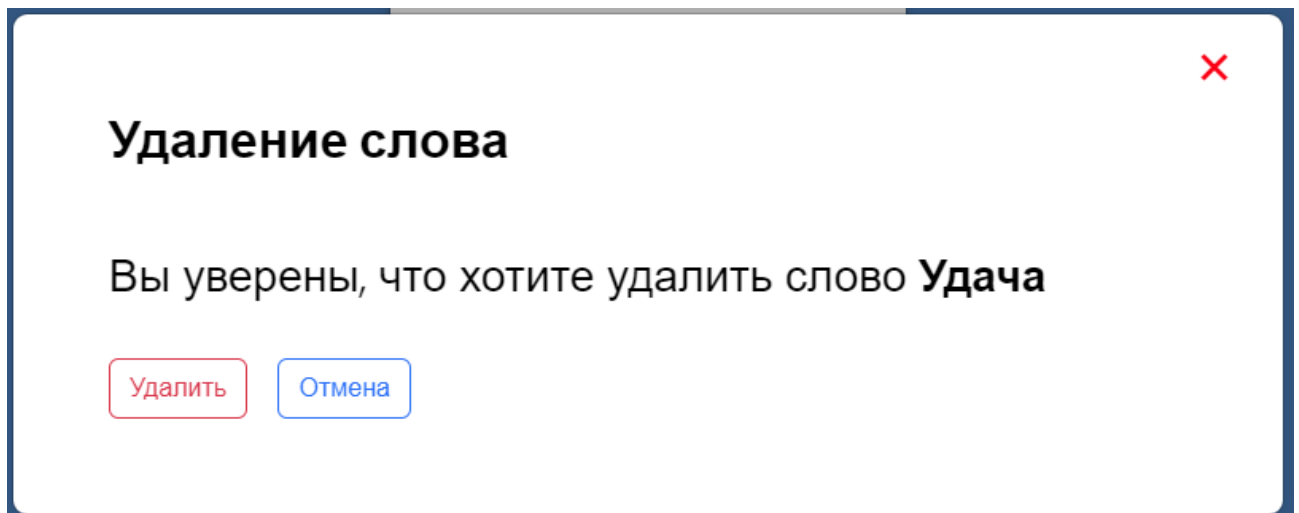


Рисунок Г.9 – Форма модального окна удаления слова

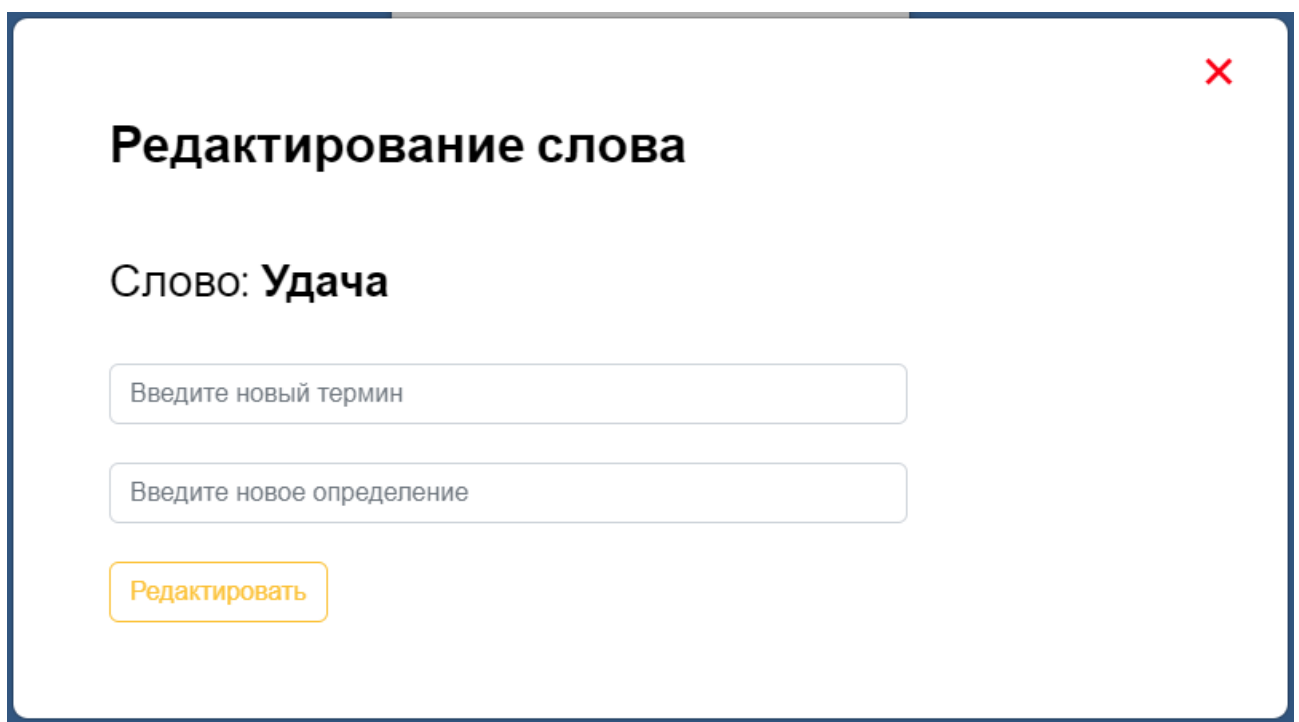


Рисунок Г.10 – Форма модального окна редактирования слова

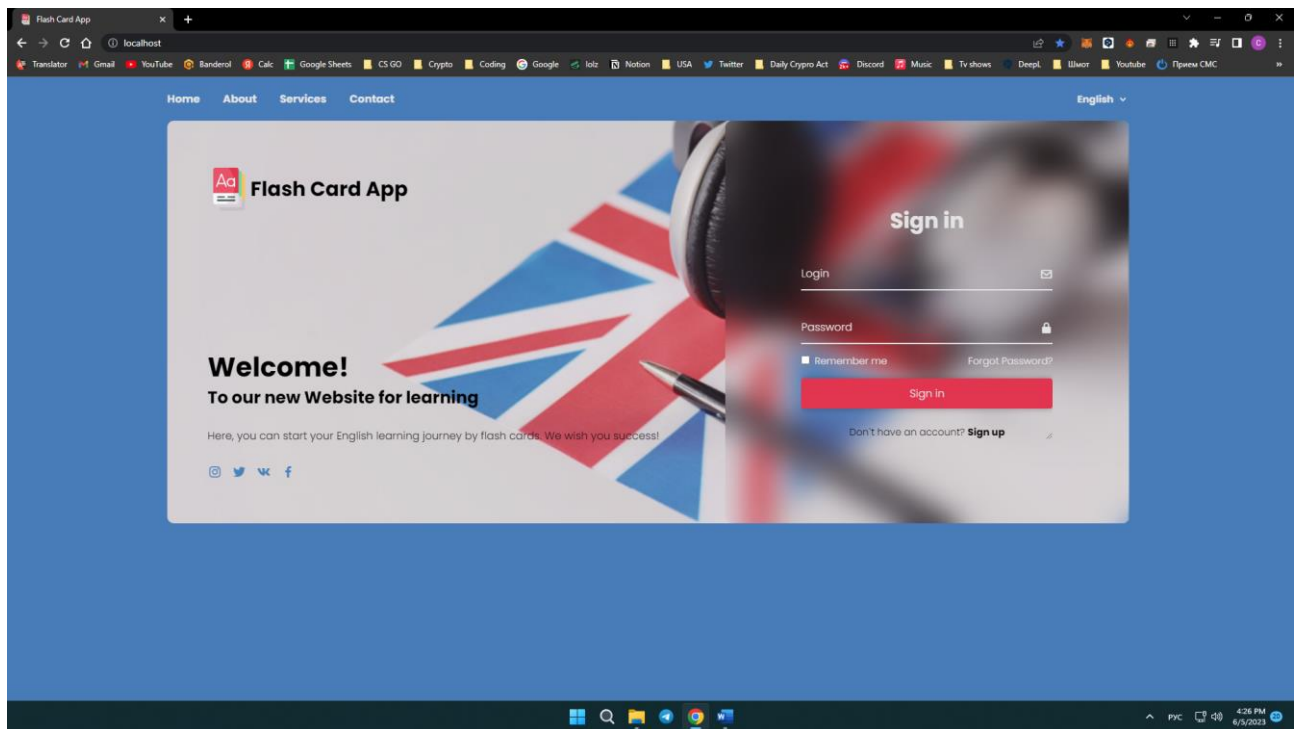


Рисунок Г.11 – Отображение web приложения в Google Chrome

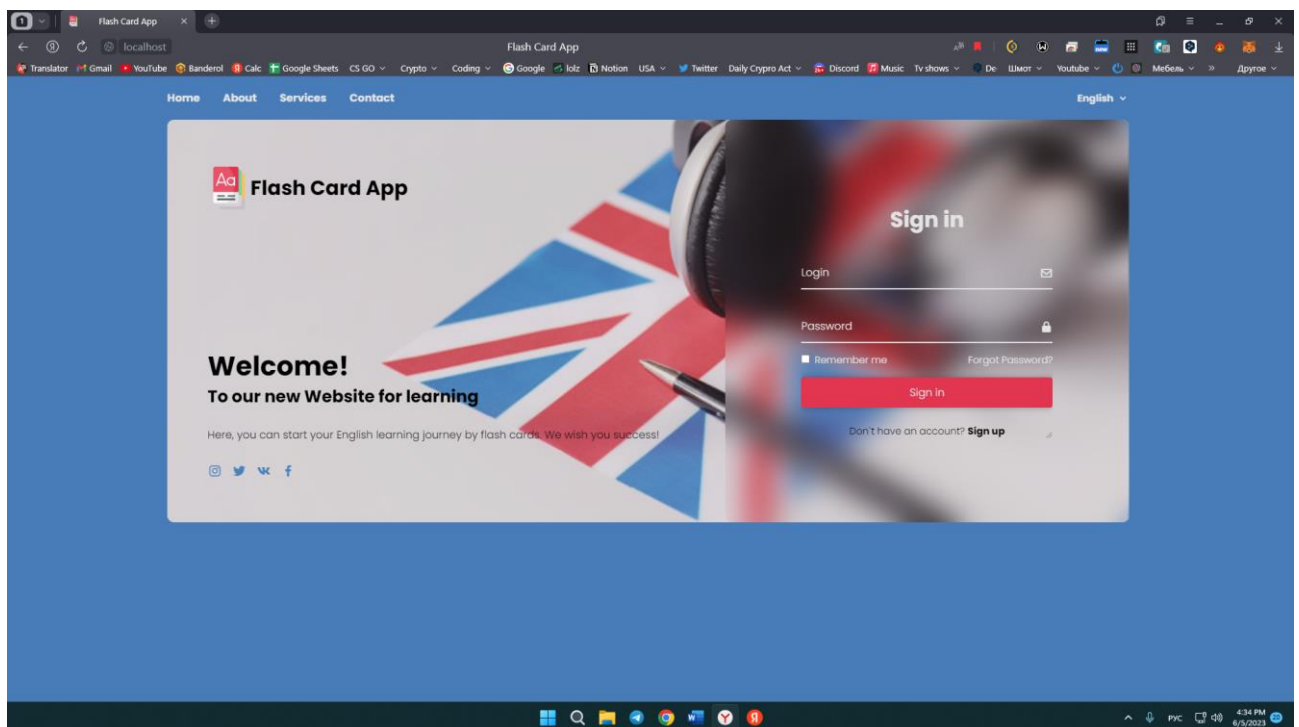


Рисунок Г.12 – Отображение web приложения в Yandex

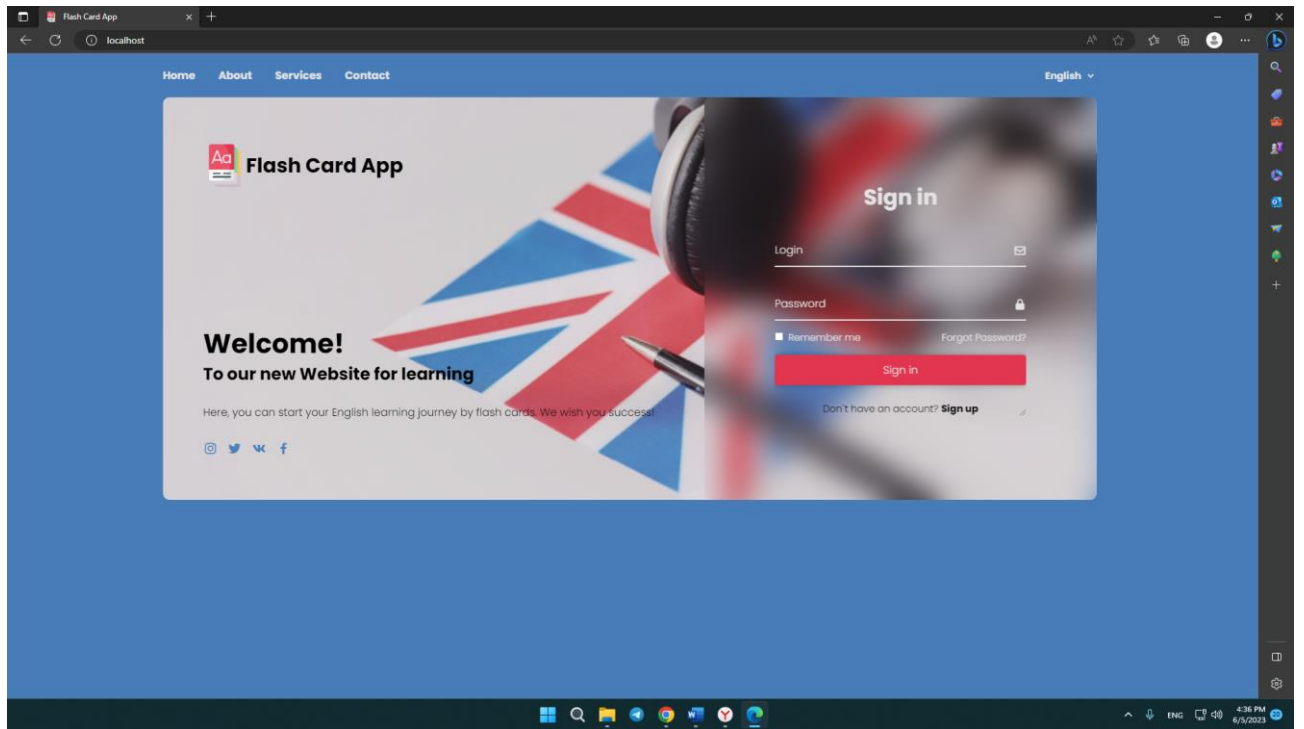


Рисунок Г.13 – Отображение web приложения в Microsoft Edge

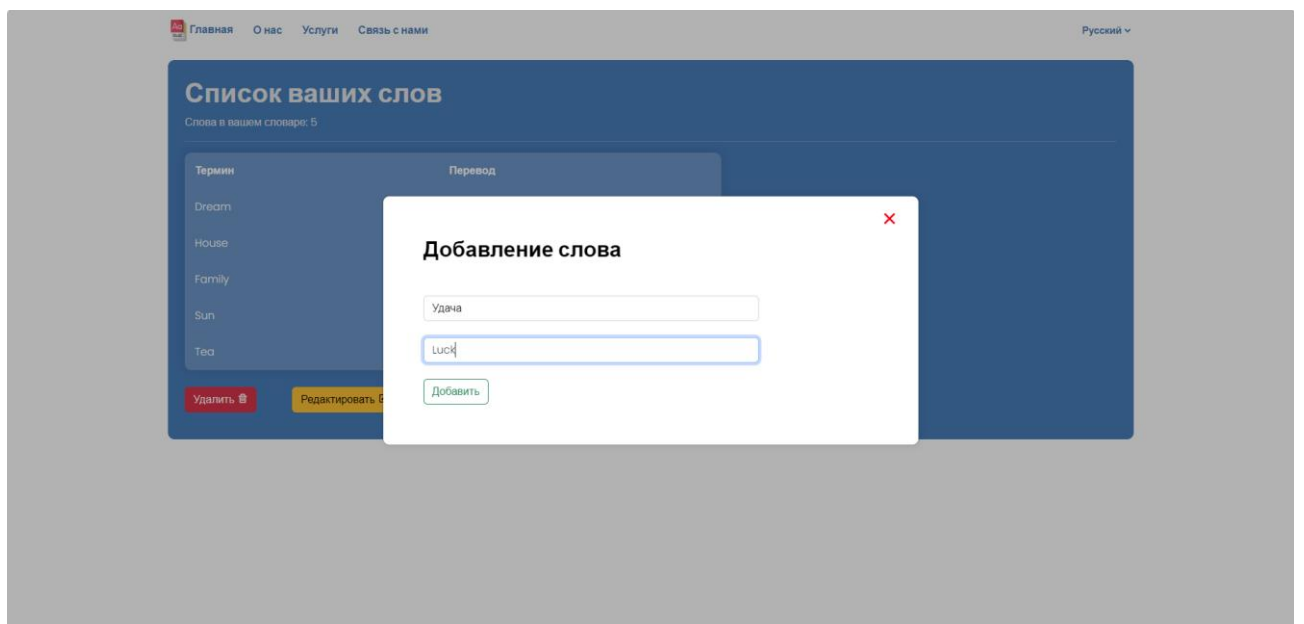


Рисунок Г.14 – Добавление новой записи в базу данных

st9903_11.words_Test: 6 rows total (approximately) » Next ⬇ Show all

id	word	definition
1	Dream	Мечта
2	House	Дом
3	Family	Семья
5	Sun	Солнце
7	Tea	Чай
8	Удача	Luck

Рисунок Г.15 – Успешное добавление записи в базу данных

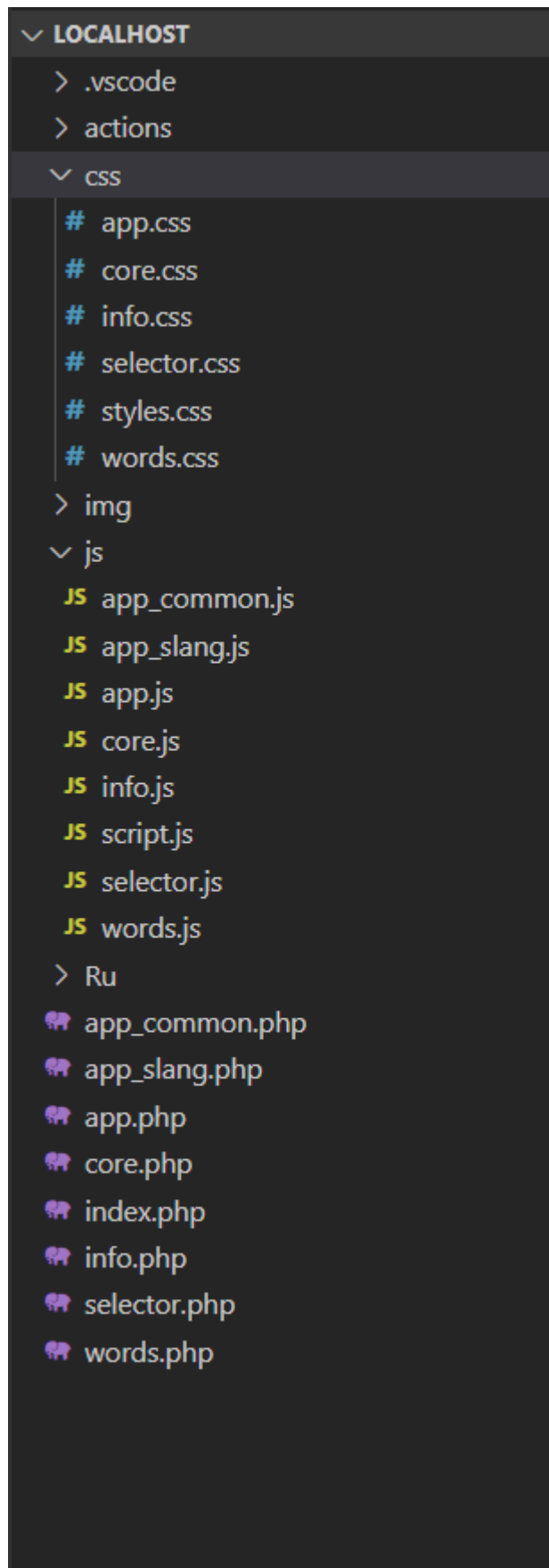


Рисунок Г.16 – Файловая структура курсового проекта

Приложение Д
(обязательное)

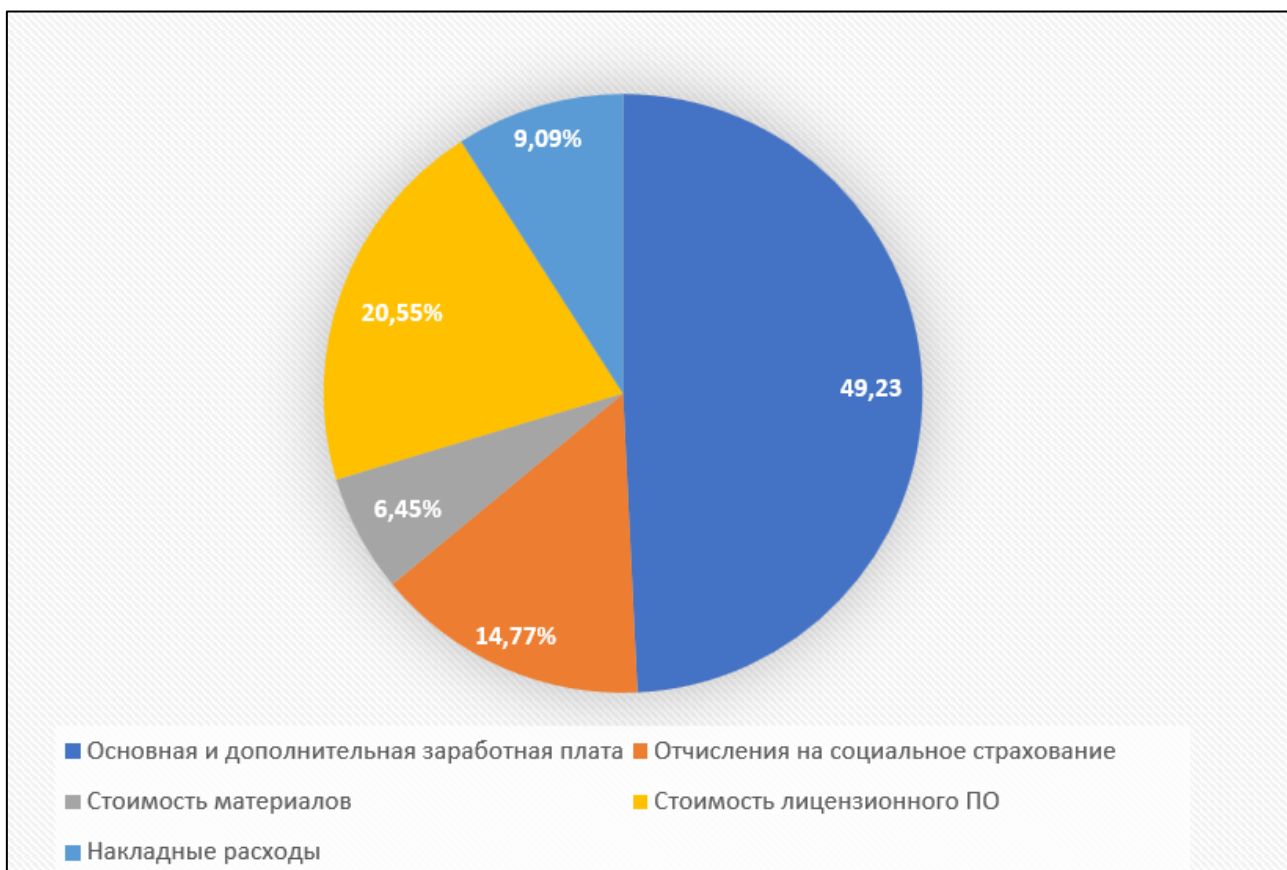


Рисунок Д.1 – Диаграмма расходов на программный продукт