



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ
КАФЕДРА

Информатика и системы управления
Программное обеспечение ЭВМ и информационные технологии

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОМУ ПРОЕКТУ

НА ТЕМУ:

**веб-приложение популярного туристического
направления**

Студент ИУ7-65Б
(Группа)

(Подпись, дата)

Нгуен Ф. С.
(И. О. Фамилия)

Руководитель курсового проекта

(Подпись, дата)

Русакова З.Н.
(И. О. Фамилия)



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

УТВЕРЖДАЮ

Заведующий кафедрой ИУ7

(Индекс)

И.В.Рудаков

(И.О.Фамилия)

« ____ » _____ 2021 г.

З А Д А Н И Е

на выполнение курсового проекта

по дисциплине Базы данных

Веб-приложение популярного туристического направления

(Тема курсового проекта)

Студент Нгуен Ф. С. гр. ИУ7-66Б

(Фамилия, инициалы, индекс группы)

График выполнения проекта: 25% к 4 нед., 50% к 7 нед., 75% к 11 нед., 100% к 14 нед.

1. Техническое задание

Спроектировать и разработать веб-приложение популярного туристического направления с возможностью поиска (фильтра) по городам и по категориям, отметки любимых мест, оценки и комментариев собственного опыта. Спроектировать и реализовать базу данных.

2. Оформление курсового проекта

2.1. Расчетно-пояснительная записка на 25-30 листах формата А4.

Расчетно-пояснительная записка должна содержать постановку введение, аналитическую часть, конструкторскую часть, технологическую часть, экспериментально-исследовательский раздел, заключение, список литературы, приложения.

2.2. Перечень графического материала (плакаты, схемы, чертежи и т.п.) На защиту проекта должна быть представлена презентация, состоящая из 15-20 слайдов. На слайдах должны быть отражены: постановка задачи, использованные методы и алгоритмы, расчетные соотношения, структура комплекса программ, диаграмма классов, интерфейс, характеристики разработанного ПО, результаты проведенных исследований.

Дата выдачи задания « ____ » _____ 20__ г.

Руководитель курсового проекта

(Подпись, дата)

Русакова З.Н.

(И.О.Фамилия)

Студент

(Подпись, дата)

Нгуен Ф. С.

(И.О.Фамилия)

Оглавление

Введение	5
1. Аналитическая часть.....	6
1.1. Постановка задачи	6
1.2. Формализация данных	6
1.3. Анализ моделей баз данных.....	6
1.3.1. Основные функции СУБД	7
1.3.2. Классификация СУБД по модели данных	7
1.4. Выводы из аналитического раздела	9
2. Конструкторская часть	10
2.1. Сценарий использования	10
2.2. Проектирование базы данных	10
2.3. Проектирование процедур и триггеров.....	12
2.4. Вывод	13
3. Технологическая часть.....	14
3.1. Архитектура программного обеспечения	14
3.2. Выбор среды и языка разработки серверной части:	14
3.3. Используемые инструменты и технологии веб-приложения	16
3.4. Реализация	16
3.5. Интерфейс приложения.....	16
3.6. Вывод	17
4. Экспериментальная часть.....	18
Заключение.....	21
Список используемой литературы	22

Введение

В современном мире множество ресторанов, аттракционов, развлекательных заведений, обслуживающих миллионы посетителей. Очень удобно отслеживать информацию, а также отзывы о достопримечательностях во время путешествий, когда их наличие в одном приложении разное.

Цель данной работы – реализовать веб-приложение популярного туристического направления с возможностью фильтра по городам и по категориям любимых мест, оценки и комментариев собственного опыта.

Для достижения поставленной цели требуется решить следующие задачи:

1. выбрать язык программирования, фреймворк для создания веб-сервера;
2. выбрать CSS библиотеку для разработки адаптивного интерфейса;
3. выбрать базу данных для хранения информации и API (Application Programming Interface) для выбранной базы данных;
4. спроектировать и разработать базу данных;
5. спроектировать и разработать веб-приложение с помощью выбранных технологий;
6. разработать гибкий интерфейс для созданного веб-приложения;

1. Аналитическая часть

В данном разделе будет проанализирована поставленная задача и рассмотрены различные способы ее реализации.

1.1. Постановка задачи

Необходимо спроектировать и разработать веб-приложение популярного туристического направления с возможностью поиска (фильтра) по городам и по категориям, отметки любимых мест, оценки и комментариев собственного опыта.

1.2. Формализация данных

В соответствии с поставленной задачей необходимо разработать клиентсерверное веб-приложение с возможностью аутентификации пользователей.

База данных должна хранить информацию о:

- Местях;
- Изображениях;
- Комментариях;
- Учетной записи пользователя.

Таблица 1. Формализация данных

Категория	Сведения
1. место	Название, тип(ресторан/ достопримечательность/итд), местоположение
2. изображение	Ссылка, тип, linkid
3. комментарий	Содержание, автор, место, дата публикации.
4. Пользователь	Имя, логин, адрес почты

1.3. Анализ моделей баз данных

Система управления базами данных (сокр. СУБД) — совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных [1].

1.3.1. Основные функции СУБД

Основными функциями СУБД являются:

- управление данными во внешней памяти;
- управление данными в оперативной памяти с использованием дискового кэша;
- журнализация изменений, резервное копирование и восстановление базы данных после сбоев;
- поддержка языков БД.

1.3.2. Классификация СУБД по модели данных

Модель данных — это абстрактное, самодостаточное, логическое определение объектов, операторов и прочих элементов, в совокупности составляющих абстрактную машину доступа к данным, с которой взаимодействует пользователь. Эти объекты позволяют моделировать структуру данных, а операторы — поведение данных. [2]

Существует 3 основных типа моделей организации данных:

- иерархическая;
- сетевая;
- реляционная.

В иерархической модели данных используется представление базы данных в виде древовидной структуры, состоящей из объектов различных уровней. Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня. Такие объекты находятся в отношении предка к потомку, при этом возможна ситуация, когда объект-предок имеет несколько потомков, тогда как у объекта-потомка обязателен только один предок.

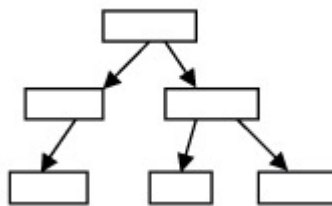


Рисунок 1. структура иерархической модели данны

В сетевой модели данных, в отличии от иерархической, у потомка может иметься любое число предков. Сетевая БД состоит из набора экземпляров определенного типа записи и набора экземпляров определенного типа связей между этими записями.

Главным недостатком сетевой модели данных являются жесткость и высокая сложность схемы базы данных, построенной на основе этой модели. Так как логика процедуры выбора данных зависит от физической организации этих данных, то эта модель не является полностью независимой от приложения. Иначе говоря, если будет необходимо изменить структуру данных, то нужно будет изменять и приложение.

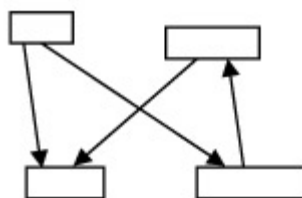


Рисунок 2. структура сетевой модели данных

Реляционная модель данных является совокупностью данных и состоит из набора двумерных таблиц. При табличной организации отсутствует иерархия элементов. Таблицы состоят из строк – записей и столбцов – полей. На пересечении строк и столбцов находятся конкретные значения. Для каждого поля определяется множество его значений. За счет возможности просмотра строк и столбцов в

любом порядке достигается гибкость выбора подмножества элементов.

Реляционная модель является удобной и наиболее широко используемой формой представления данных.

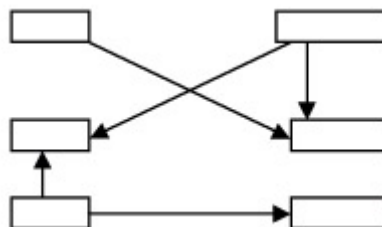


Рисунок 3. структура реляционной модели данных

Наиболее популярными реляционными СУБД являются Oracle, Microsoft SQL Server и PostgreSQL.

1.4. Выводы из аналитического раздела

В данном разделе была проанализирована поставленная задача и рассмотрены способы ее реализации. Также были рассмотрены разные типы СУБД. В качестве используемой в данной работе была выбрана реляционная СУБД Microsoft SQL Server.

2. Конструкторская часть

В данном разделе будет спроектированы база данных и приложение.

2.1.Сценарий использования

В разделе необходимо построить диаграмму прецедентов (Use Case Diagram), состоящая из графической диаграммы, описывающей действующие лица и прецеденты – конкретные действия, которые выполняет пользователь при работе с системой.

Данная диаграмма предназначена для определения функциональных требований. В системе есть три типа пользователей:

- неавторизованный (гость);
- авторизованный;
- администратор (модератор).

На рисунке представлена Use Case Diagram для действий всех трёх типов пользователей.

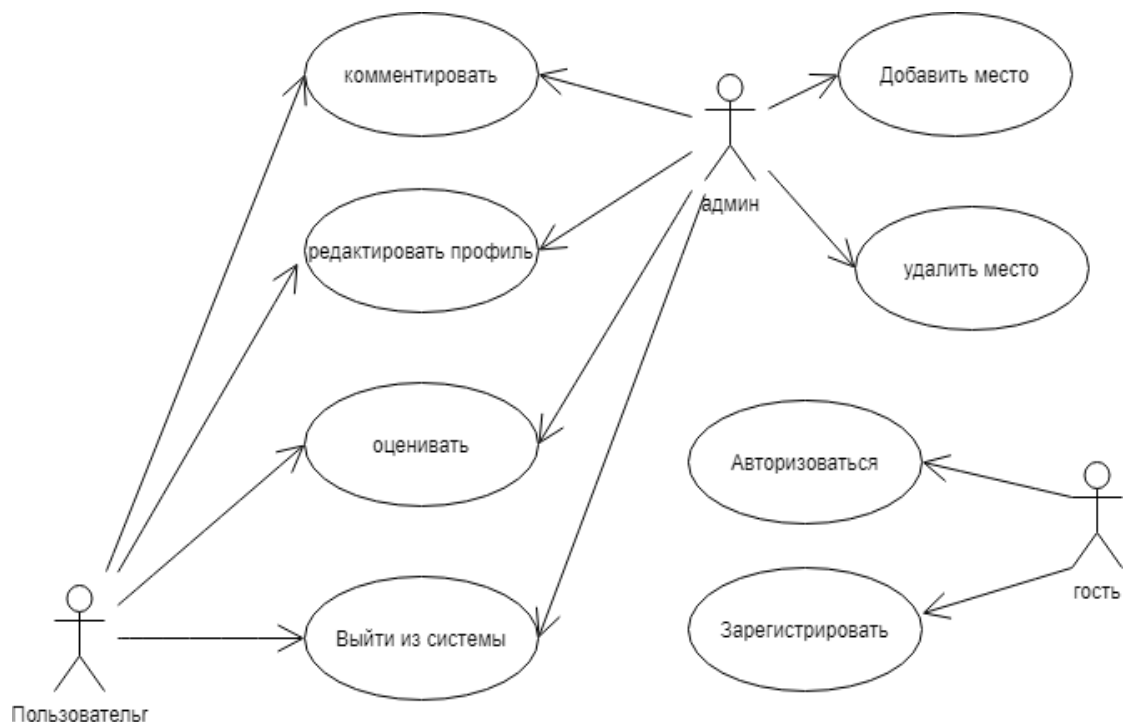


Рисунок 4. Use Case Diagram для действий всех трёх типов пользователей.

2.2.Проектирование базы данных

База данных должна хранить рассмотренные в таблице 1.1 данные. В соответствии с этой таблицей можно выделить следующие таблицы:

- User
- Place
- Comment
- Image

Таблица **Place** должна хранить информацию о местах:

- id - уникальный идентификатор пользователя, РК, uniqueidentifier;
- name – имя, varchar;
- location – адрес
- description – описание text;
- rating - рейтинг места, float;

Таблица **User** должна хранить информацию о пользователях:

- id - уникальный идентификатор пользователя, РК, uniqueidentifier;
- name – имя, varchar;
- login – логин пользователя, используется для авторизации, должен быть уникальным, nchar(20);
- password – пароль пользователя, используется для авторизации, должен храниться в хешированном состоянии, nchar(64);
- email - адрес почты, varchar, unique;
- avatar – путь до изображения, varchar

Таблица **Review** должна хранить информацию о комментариях:

- id - уникальный идентификатор комментария, РК, serial;
- place_id – идентификатор места, FK, integer;
- auth_id – идентификатор автора комментария, FK, integer;

- comment – содержание комментария, text;
- rating - рейтинг места, integer
- public_date – время публикации комментария, date;

Таблица Image должна хранить информацию о изображениях:

- id – уникальный идентификатор изображения, PK, serial;
- path – путь до изображения, varchar;
- type – тип, integer;
- placeid – идентификатор места, integer

На рисунке 6 представлена ER – модель базы данных.

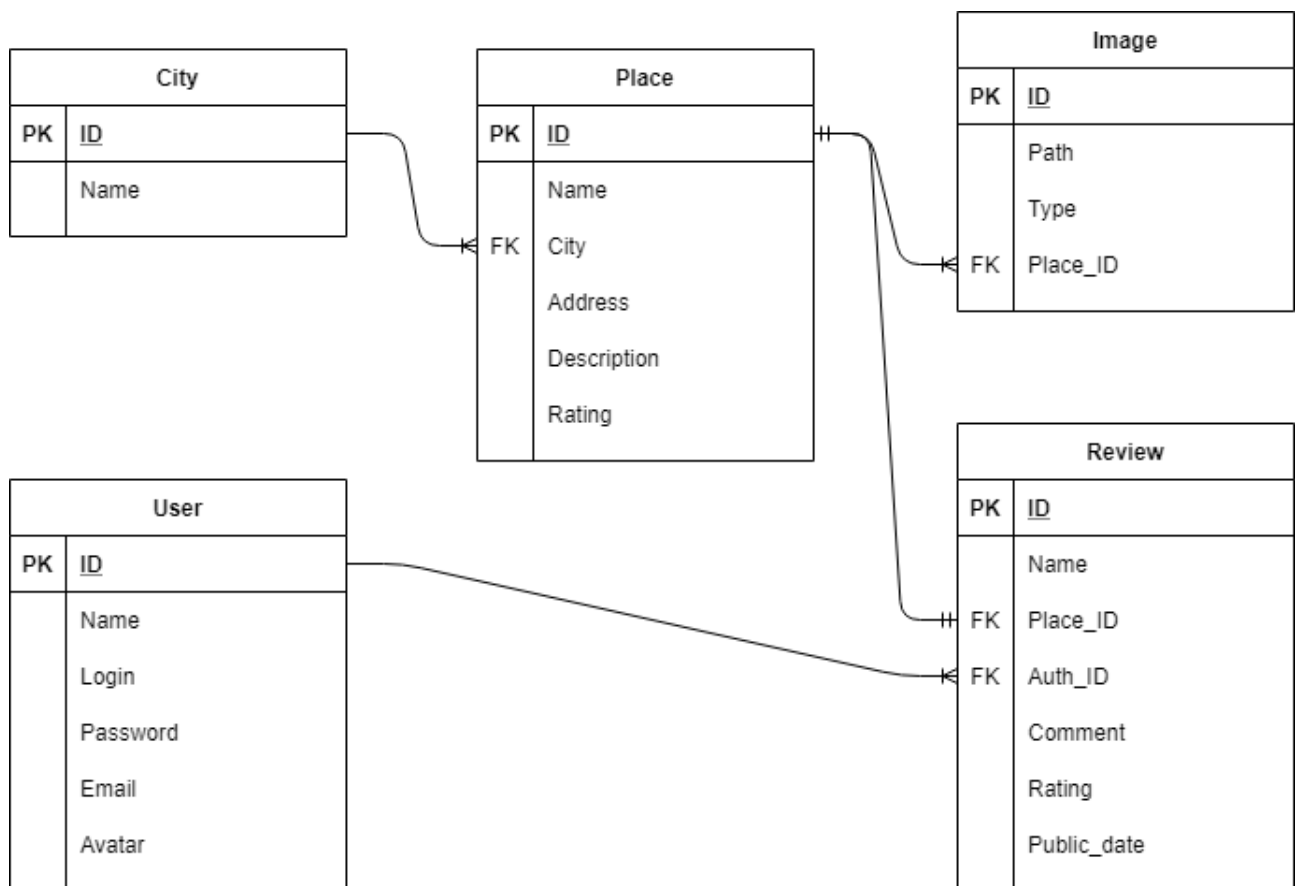


Рисунок 5. ER – модель базы данных

2.3.Проектирование процедур и триггеров

Диаграмма для функции авторизации: На Рисунке показана IDEF0 диаграмма Функции авторизации.

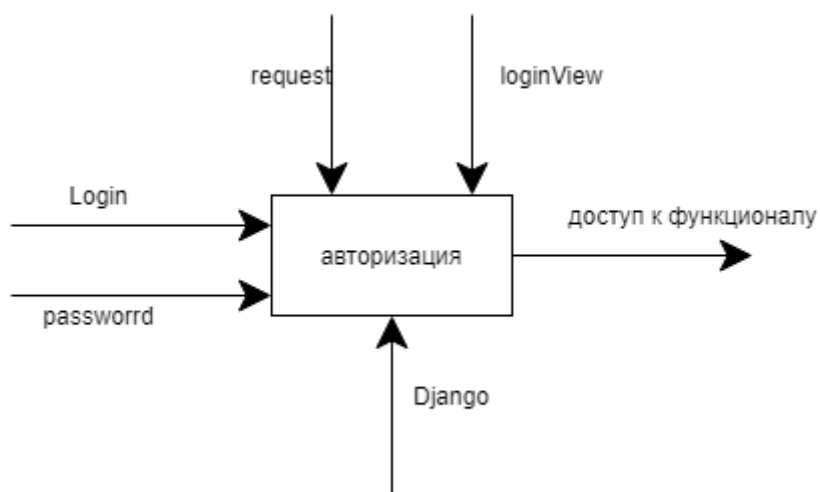


Рисунок 6. IDEF0 диаграмма Функции авторизации.

Диаграмма функции редактирования профиля: на рисунке показана IDEF0 диаграмма функции редактирования профиля:

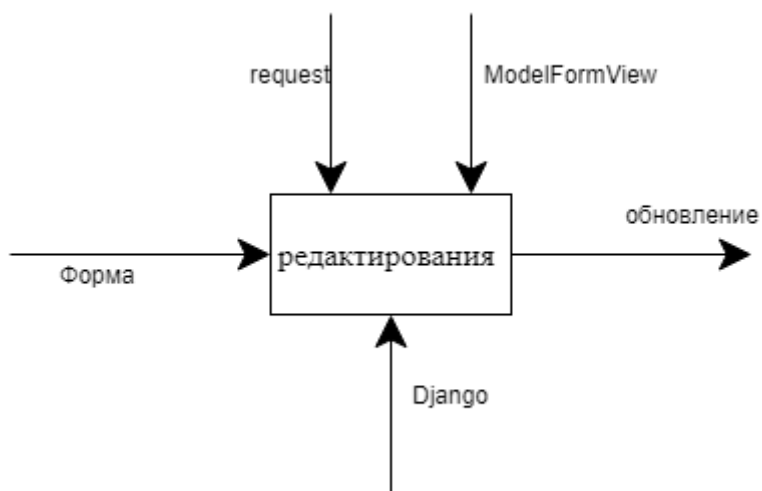


Рисунок 7. IDEF0 диаграмма функции редактирования профиля

2.4. Вывод

В данном разделе были представлены схемы и диаграммы представляющую структуру БД, а также описывающую логику ее работы. Так же были показаны диаграммы некоторых основных функций ПО.

3. Технологическая часть

В данном разделе необходимо определить технологический стек для разработки программного обеспечения.

3.1. Архитектура программного обеспечения

В соответствии с техническим заданием будущее приложение должно иметь трехзвенную систему: веб-приложение(представление), базу данных(модель) и логику на серверной части(контроллер).

Основная цель применения этой концепции состоит в отделении бизнес-логики(модели) от её визуализации. За счёт такого разделения повышается возможность повторного использования кода.

3.2. Выбор среды и языка разработки серверной части:

При выборе языка будем руководствоваться следующими факторами:

- платформ независимый;
- динамический;
- высокоуровневый;
- многопоточный.

Под динамическим в данном случае понимается, что программа может выполнять обширное количество задач во время обработки информации, которая может быть использовано для проверки и разрешения доступа к объектам на время выполнения.

Всем этим критериям удовлетворяет язык Python3. Основным достоинством является то, что с помощью данного языка можно «разбить» реализацию: интерфейс, работу с файлами и элементами формы организовать с помощью встроенных средств.

При выборе среды разработки рассматриваются и другие возможные среды, которые помогут автоматизировать процесс разработки. Для

разработки данной программы необходимо обеспечение средой следующих возможностей:

- удобные инструменты для отладки и поиска ошибок, в случае их возникновения;
- разработка более гибкой и надежной программы путем обработки различных исключительных ситуаций, возникающих в результате некорректной работы программы;
- использование всплывающих подсказок во время написания кода программы, что обеспечивает значительную экономию времени и повышения уровня продуктивности.

Среда разработки PyCharm[2] поддерживает все эти возможности, а также многие другие, такие как:

- инструменты для запуска тестов и анализа покрытия кода, включая поддержку всех популярных фреймворков для тестирования;
- инструменты для работы с базами данных и SQL файлами, включая удобный клиент и редактор для схемы базы данных;
- окно классов для перехода по исходному коду по типам, а не файлам;
- обозреватель документов для просмотра и поиска документации по продуктам на локальном компьютере или в Интернете;
- умное автодополнение, инструменты для анализа качества кода, удобная навигация, расширенные рефакторинги и форматирование;
- окно Свойства для настройки свойств и событий элементов управления в пользовательском интерфейсе.

Среду разработки PyCharm стоит рассматривать как интеллектуальную среду разработки, понимающую код. В процессе его

написания программистом она занимается построением синтаксического дерева, определением особенностей размещенных ссылок, анализом возможных путей исполнения операторов и передачи данных.

3.3. Используемые инструменты и технологии веб-приложения

Для облегчения разработки веб-приложения нужно использовать фреймворки, которые являются набором шаблонов или заготовок, облегчающий разработку и объединение разных модулей программного проекта.

Одними из популярных фреймворков являются PHP-фреймворки Zend Framework и Symfony или Django[5], написанный на Python.

Основным критерием выбора является язык разработки серверной части, который должен корректно взаимодействовать с Фреймворком.

Анализируя различные инструменты, для разработки веб-приложения был выбран универсальный фреймворк с открытым исходным кодом Django написанный на Python3[7]. Для написания лицевой части приложения были использованы HTML[3], CSS[6], JavaScript, Bootstrap[4]. В качестве СУБД была выбрана SQLite[1].

3.4. Реализация

Для работы с базой данных Django[5] использует собственную технологию программирования, в которой модель данных описывается классами Python, и по ней генерируется схема базы данных. В качестве среды разработки выбран редактор PyCharm.

В терминале PyCharm с помощью команды `python manage.py runserver` запускается сервер, выделяется адрес для localhost. В данном случае был выделен адрес 127.0.0.1:8000.

3.5. Интерфейс приложения

Ниже представлен интерфейс полученного веб-приложения.

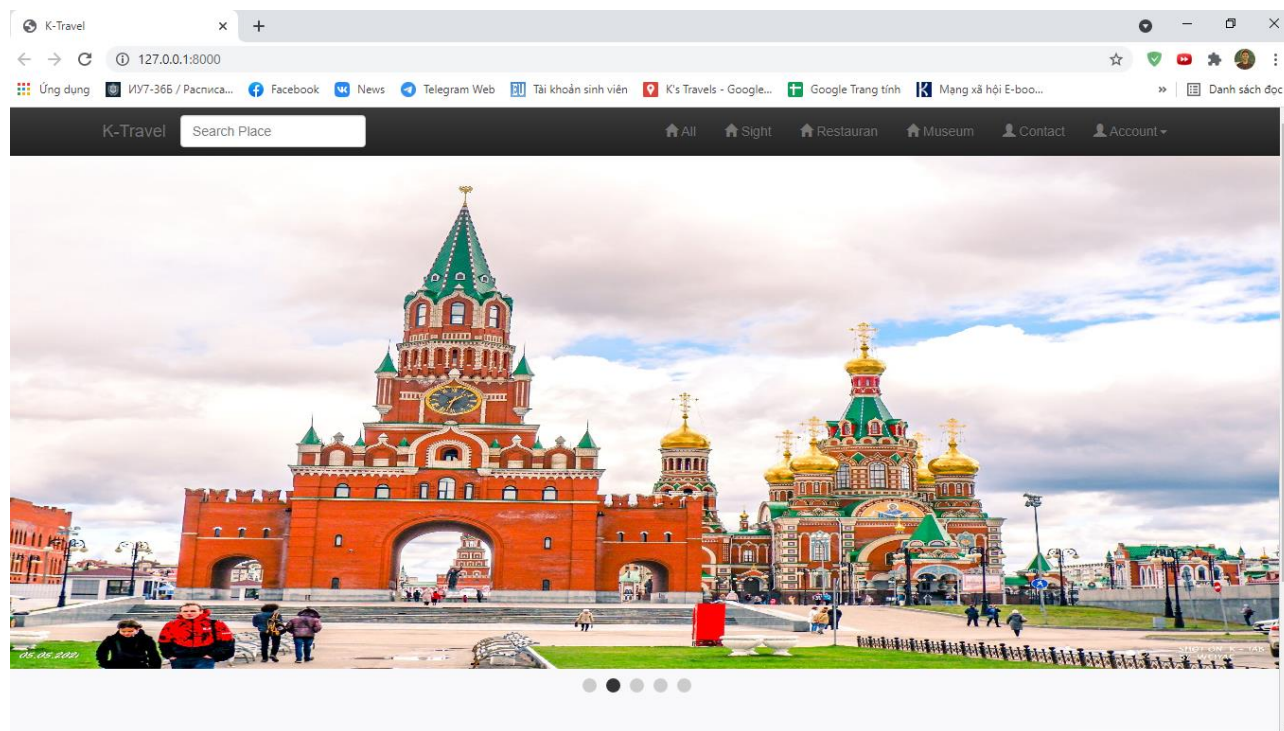


Рисунок 8. Интерфейс приложения

3.6. Вывод

В данном разделе было описано и обосновано выбор языка и средств реализации дпнного проекта. Был продемонстрировано внутренний интерфейс программы.

4. Экспериментальная часть

В рамках данного проекта были проведены эксперименты, описанные ниже:

- проверка на корректное создание, удаление, обновление данных в БД;
- корректная обработка в случае, если данные в бд уже существуют.

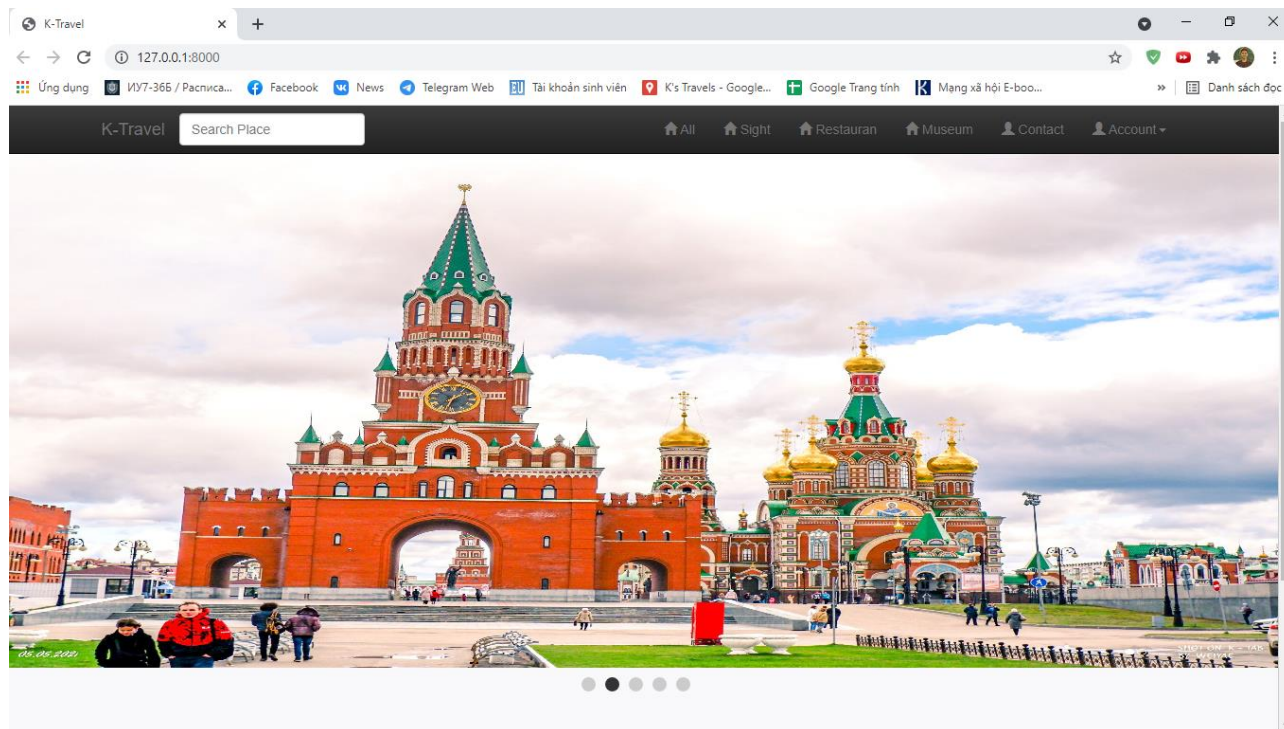


Рисунок 9. Интерфейс главного меню

Страница "All Place" показывает все местоположения.

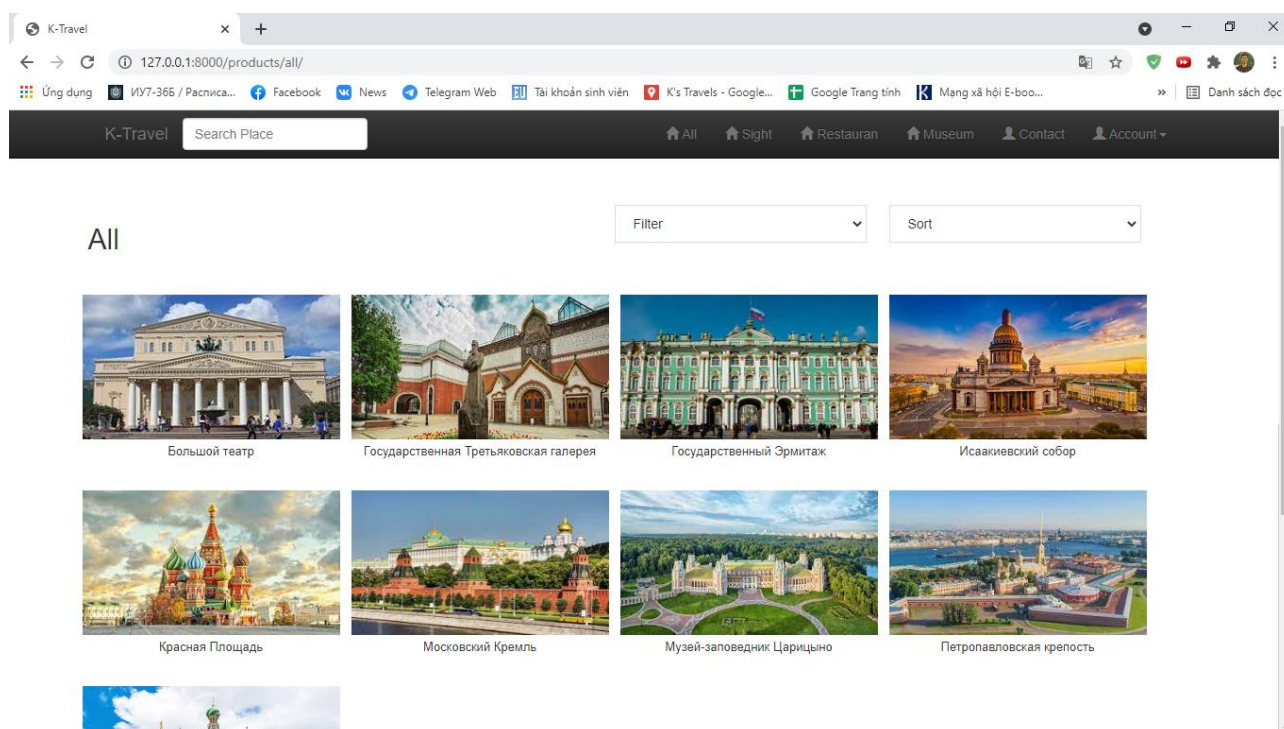


Рисунок 10. Страница "All Place"

На странице представлена информация по местоположению

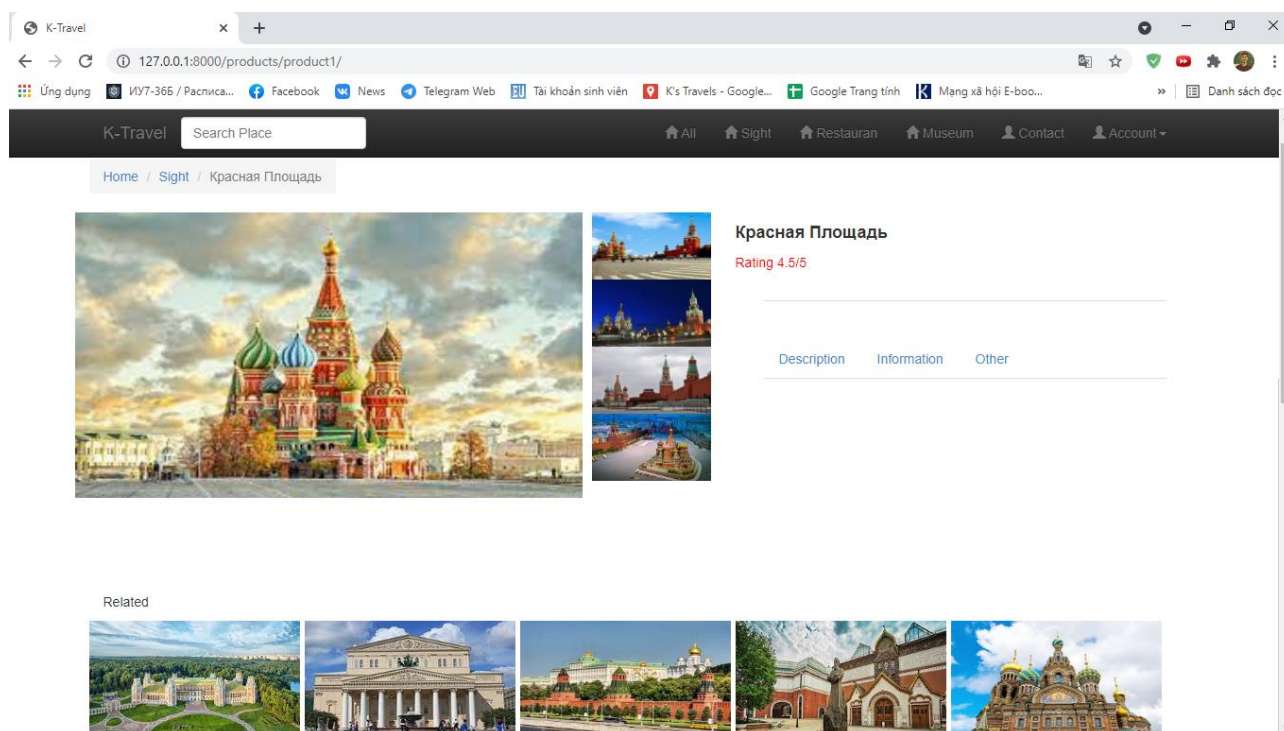


Рисунок 11. информация по местоположению

На рисунке 12 представлена функция поиска по названию

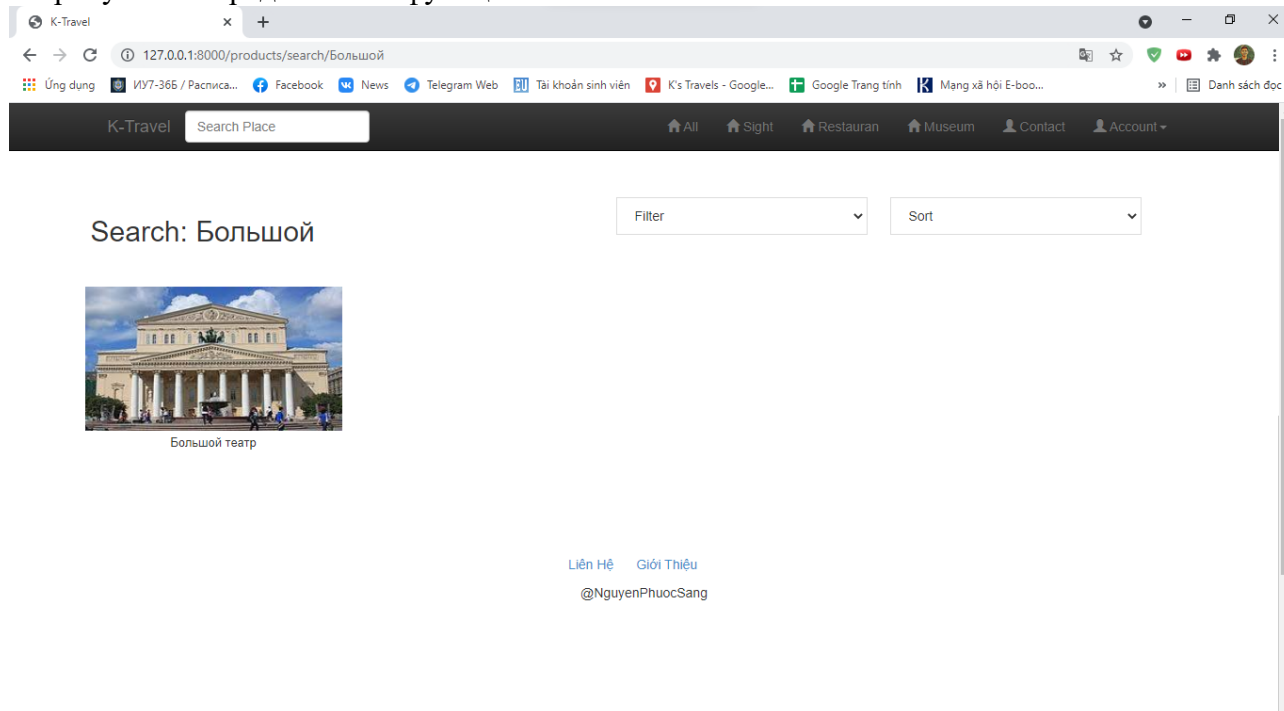
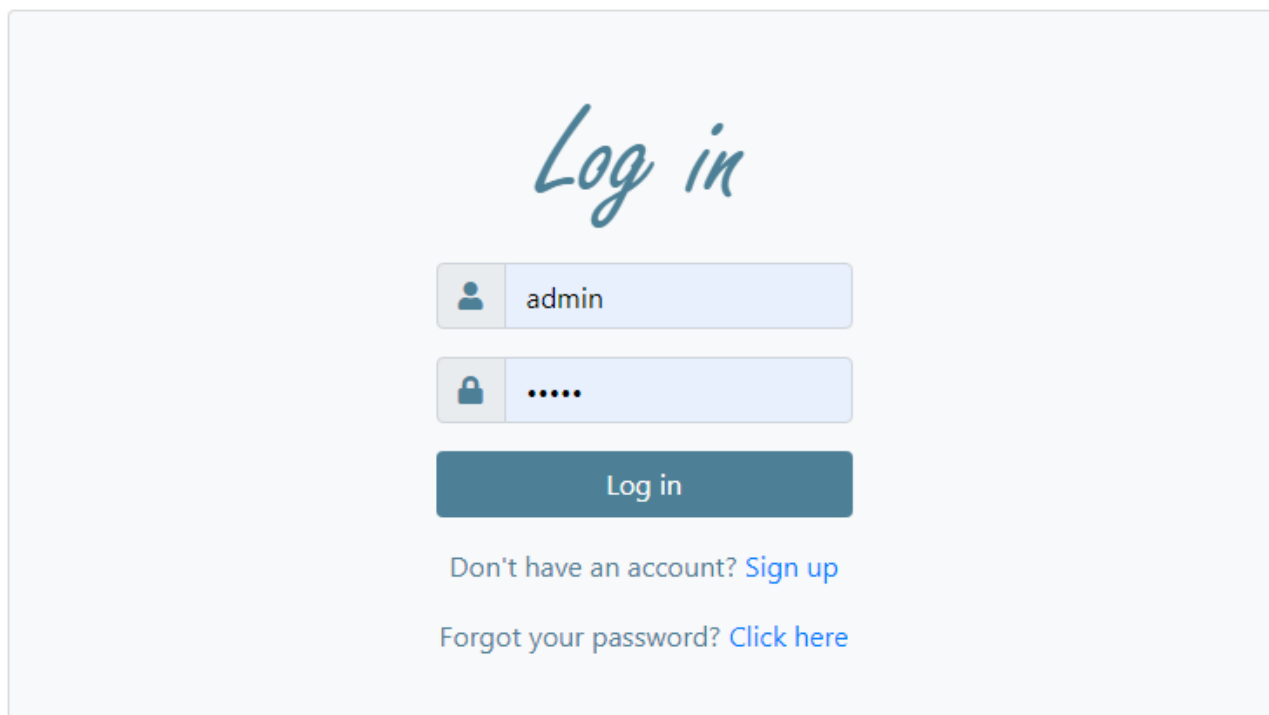


Рисунок 12 функция поиска по названию

На рисунке 13 показано модуль Авторизации, где пользователю необходимо ввести логин и пароль от системы. В случае успеха, пользователю будет доступен функционал в соответствии с его ролью в приложении.



The login form features a large, stylized 'Log in' title at the top. Below it are two input fields: the first for the username 'admin' and the second for a password represented by six dots. A dark blue 'Log in' button is positioned below the password field. At the bottom, there are two links: 'Don't have an account? Sign up' and 'Forgot your password? Click here'.

Log in

admin

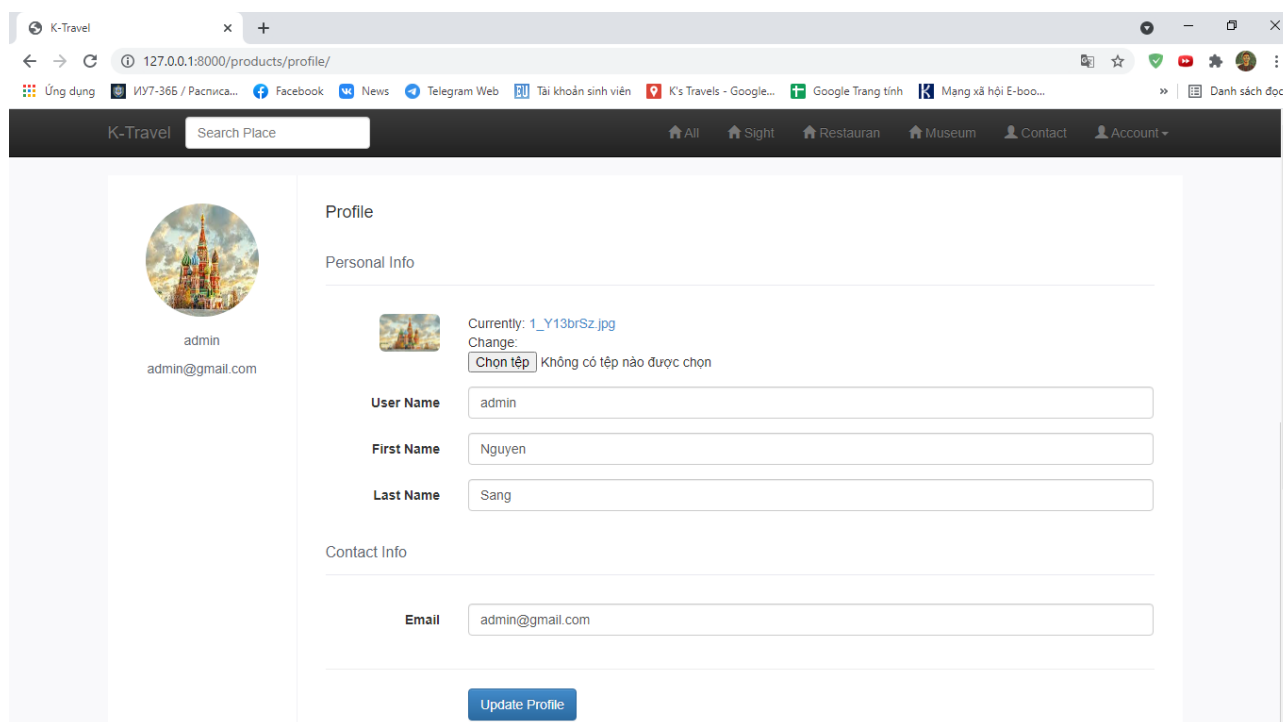
.....

Log in

Don't have an account? [Sign up](#)

Forgot your password? [Click here](#)

Рисунок 13. модуль Авторизации



The profile page is displayed in a web browser window. On the left is a sidebar with a circular profile picture of a cathedral, the username 'admin', and the email 'admin@gmail.com'. The main content area is titled 'Profile' and contains two sections: 'Personal Info' and 'Contact Info'. The 'Personal Info' section includes a profile picture change button, a 'User Name' field with 'admin', and 'First Name' (Nguyen) and 'Last Name' (Sang) fields. The 'Contact Info' section has an 'Email' field with 'admin@gmail.com'. An 'Update Profile' button is at the bottom.

Profile

Personal Info

Currently: 1_Y13brSz.jpg
Change:
 Không có tệp nào được chọn

User Name

First Name

Last Name

Contact Info

Email

Рисунок 14. страница профиля

Заключение

В ходе проекта разработаны базы данных для хранения информации, гибкий интерфейс, а так же веб-приложение с помощью выбранных технологий;

Список используемой литературы

1. HTML, CSS, JAVASCRIPT Tutorial [www.w3schools.com]
2. Django documentation [<https://docs.djangoproject.com/>]
3. [<https://stackoverflow.com/>]