**ОТЧЕТ**по учебной практике  
ПМ.07 СОАДМИНИСТРИРОВАНИЕ БАЗ ДАННЫХ И СЕРВЕРОВ  
Специальность 09.02.07 Информационные системы и программирование

Выполнил  
студент гр. 494 \_\_\_\_\_\_\_\_\_\_ В.Е. Марков

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватели УП \_\_\_\_\_\_\_\_\_\_ А.В. Фомин

Оглавление

[ВВЕДЕНИЕ 3](#_Toc129688626)

[1. АНАЛИТИЧЕСКАЯ ЧАСТЬ 4](#_Toc129688627)

[1.1. Анализ методов решений 4](#_Toc129688628)

[1.2. Обзор средств разработки 5](#_Toc129688629)

[1.2.1. Среда разработки 5](#_Toc129688630)

[1.2.2. Система управления базами данных 5](#_Toc129688631)

[1.3. Обоснование выбора 7](#_Toc129688632)

[1.3.1. Средства разработки 7](#_Toc129688633)

[1.3.2. Языки программирования 7](#_Toc129688634)

[1.3.3. Система управления базами данных 7](#_Toc129688635)

[1.3.4. Дополнительные инструменты 8](#_Toc129688636)

[ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ 8](#_Toc129688637)

[2.1. Структура базы данных 8](#_Toc129688638)

[2.1.1. Схема базы данных 8](#_Toc129688639)

[2.1.2. Структура таблицы Country 9](#_Toc129688640)

[2.1.3. Структура таблицы User 10](#_Toc129688641)

[2.1.4. Структура таблицы History 11](#_Toc129688642)

[2.2. Описание функций 12](#_Toc129688643)

[2.2.1 Описания функций 12](#_Toc129688644)

[2.5. Макет интерфейса приложения 13](#_Toc129688645)

[2.6. Реализация интерфейса приложения 15](#_Toc129688646)

# ВВЕДЕНИЕ

При выполнении курсовой работы решаются задачи закрепления теоретических знаний, а также практических навыков проектирования и разработки баз данных, полученных при изучении МДК07.01 «Управление и автоматизация баз данных».

Курсовой проект представляет комплексную проектную, практическую разработку, которая в дальнейшем может стать основой выпускного дипломного проекта.

Цель курсовой работы – формирование практических навыков по проектированию и реализации базы данных для выбранной предметной области на основе полученных теоретических знаний.

Предметная область: «Служба по определению страны по IP-адресу». База данных «Служба по определению страны по IP-адресу» хранит информацию о странах, сетях и пользователях.

# 1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

## 1.1. Анализ методов решений

Необходимую систему можно реализовать несколькими способами:

1. В виде мобильного приложения.

Преимущества:

* Высокий уровень персонализации
* Возможность работы оффлайн
* Более высокая производительность и эффективность кода.

Минусы:

* Стоимость разработки и поддержки.

2. В виде web-приложения.

Преимущества:

* Простота доступа к приложению
* Не требует установки
* Простота развертывания.

Недостатки:

* Отсутствие доступа к телефону пользователя
* Зависит от интернета, работоспособности и сервера.

3. В виде настольного приложения.

Преимущества:

* Более удобны, отзывчивы, настраиваемы
* Стабильность
* Более высокая производительность и эффективность кода.

Недостатки:

* Сложность обновления
* Доступность только на устройстве пользователя
* Меньшая удобность использования.

## 1.2. Обзор средств разработки

### 1.2.1. Среда разработки

Для реализации пользовательского интерфейса был выбран Flask.

Flask ­- фреймворк для создания веб-приложений на языке программирования Python, использующий набор инструментов Werkzeug, а также шаблонизатор Jinja2. Относится к категории так называемых микрофреймворков - минималистичных каркасов веб-приложений, сознательно предоставляющих лишь самые базовые возможности.

### 1.2.2. Система управления базами данных

Microsoft SQL Server – система управления реляционными базами данных, разработанная корпорацией Microsoft. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

SQL – это инструмент, с помощью которого человек управляет базой данных. При этом ключевыми операциями являются создание таблиц, добавление записей в таблицы, изменение и удаление записей, выборка записей из таблиц, изменение структуры таблиц.

Изначально SQL был основным способом работы пользователя с базой данных и позволял выполнять следующий набор операций:

* Создание в базе данных новой таблицы;
* Добавление в таблицу новых записей;
* Изменение записей;
* Удаление записей;
* Выборка записей из одной или нескольких таблиц (в соответствии с заданным условием);
* Изменение структур таблиц.

PostgreSQL — это объектно-реляционная система управления базами данных (ОРСУБД), основанная на POSTGRES, Version 4.2.

Существует в реализациях для множества UNIX-подобных платформ, включая AIX, различные BSD-системы, HPUX, IRIX, Linux, macOS, Solaris/OpenSolaris, Tru64, QNX, а также для Microsoft Windows.

PostgreSQL создана на основе некоммерческой СУБД Postgres, разработанной как open-source проект в Калифорнийском университете в Беркли.

PostgreSQL — СУБД с открытым исходным кодом. Она поддерживает большую часть стандарта SQL и предлагает множество современных функций:

* сложные запросы;
* внешние ключи;
* триггеры;
* изменяемые представления;
* транзакционная целостность;

Кроме того, пользователи могут всячески расширять возможности PostgreSQL, например создавая свои

* типы данных;
* функции;
* операторы
* агрегатные функции;
* методы индексирования;
* процедурные языки.

Благодаря свободной лицензии, PostgreSQL разрешается бесплатно использовать, изменять и распространять всем и для любых целей — личных, коммерческих или учебных.

## 1.3. Обоснование выбора

### 1.3.1. Средства разработки

Для создания пользовательского интерфейса используется Microsoft Visual Studio Code, легкий текстовый редактор для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git.

### 1.3.2. Языки программирования

Язык, с помощью которого пишется программа, выбран язык программирования Python.

Высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Язык является полностью объектно-ориентированным в том плане, что всё является объектами. Необычной особенностью языка является выделение блоков кода пробельными отступами. Синтаксис ядра языка минималистичен, за счёт чего на практике редко возникает необходимость обращаться к документации. Сам же язык известен как интерпретируемый и используется в том числе для написания скриптов.

### 1.3.3. Система управления базами данных

Для достижения поставленной цели выбрана СУБД PostgreSQL. Данная СУБД имеет ряд преимуществ по сравнению с другими современными аналогами. К этим преимуществам относятся:

* Неограниченный размер для баз данных;
* Высокая степень защиты транзакций;
* Поддержка огромного количества модулей расширения;
* Наличие наследования;
* Легкая расширяемость.

### 1.3.4. Дополнительные инструменты

Свободный кроссплатформенный редактор диаграмм, часть GNOME Office, но может быть установлен независимо. Он может быть использован для создания различных видов диаграмм: блок-схем алгоритмов программ, древовидных схем, статических структур UML, баз данных, диаграмм сущность-связь, радиоэлектронных элементов, потоковых диаграмм, сетевых диаграмм и других.

# ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ

## 2.1. Структура базы данных

### 2.1.1. Схема базы данных

На рисунке 1 показана структура базы данных.

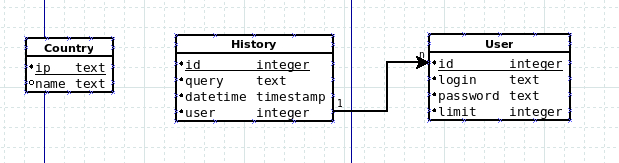


Рисунок 1- структура базы данных

База данных состоит из таблиц, размешенных в схеме:

1. Country – содержит информацию о странах и сетях в них
2. History – содержит историю запросов
3. User – содержит информацию о пользователях

### 2.1.2. Структура таблицы Country

Словарь данных для таблицы Country представлен на скриншоте и в таблице 1.

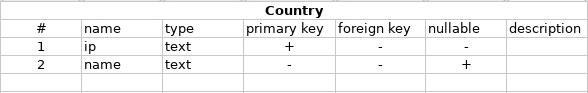


Таблица 1 – Описание полей таблицы

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Name | Type | Primary  Key | Foreign Key | Nullable | Description |
| 1 | Ip | Text | Yes | - | - |  | |
| 2 | Name | Text | - | - | Yes |  | |

Таблица Country содержит 2 столбца:

* ip – Уникальный идентификатор для метрики;
* name – Название для праздника/события.

На рисунке 2 представлен пример данных в таблице Country.



Рисунок 2 - пример данных в таблице Country

### 2.1.3. Структура таблицы User

Словарь данных для таблицы User представлен на скриншоте в таблице 2.

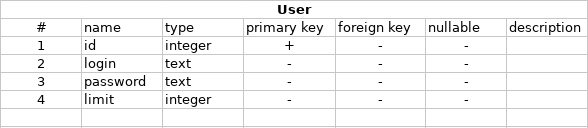


Таблица 2 – Описание полей таблицы

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Name | Type | Primary  Key | Foreign Key | Nullable | Description |
| 1 | Id | Integer | Yes | - | - |  | |
| 2 | Login | Text | - | - | - |  | |
| 3 | Password | Text | - | - | - |  | |
| 4 | Limit | Integer | - | - | - |  | |

Таблица User содержит 4 столбца:

* id – Уникальный идентификатор для метрики;
* login – Название для праздника/события;
* password – дата начала праздника/события;
* limit – дата окончания праздника/события.

На рисунке 3 представлен пример данных в таблице Holiday

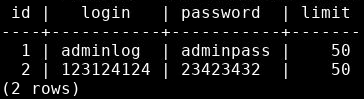


Рисунок 3 - пример данных в таблице User

### 2.1.4. Структура таблицы History

Словарь данных для таблицы History представлен в таблице 3.

Таблица 3 – Описание полей таблицы

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Name | Type | Primary  Key | Foreign Key | Unique | Description |
| 1 | Id | Integer | Yes | - | Yes |  | |
| 2 | Query | Text | - | - | - |  | |
| 3 | Datetime | Timestamp | - | - | - |  | |
| 4 | User | Integer | - | User.Id | - |  | |

Таблица Delete\_Events содержит 5 столбцов:

* id – Уникальный идентификатор записи;
* Query – Текст запроса;
* Datetime – время выполнения запроса;
* User – пользователь выполнивший запрос;

## 2.2. Описание функций

### 2.2.1 Описания функций

Создание функции api.send\_address представлено на рисунке 4

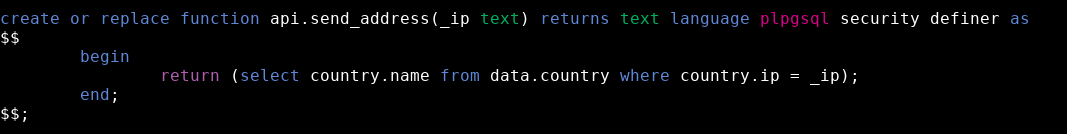


Рисунок 4 - Создание функций api.send\_address

Создание функций api.reg, api.log представлено на рисунке 5

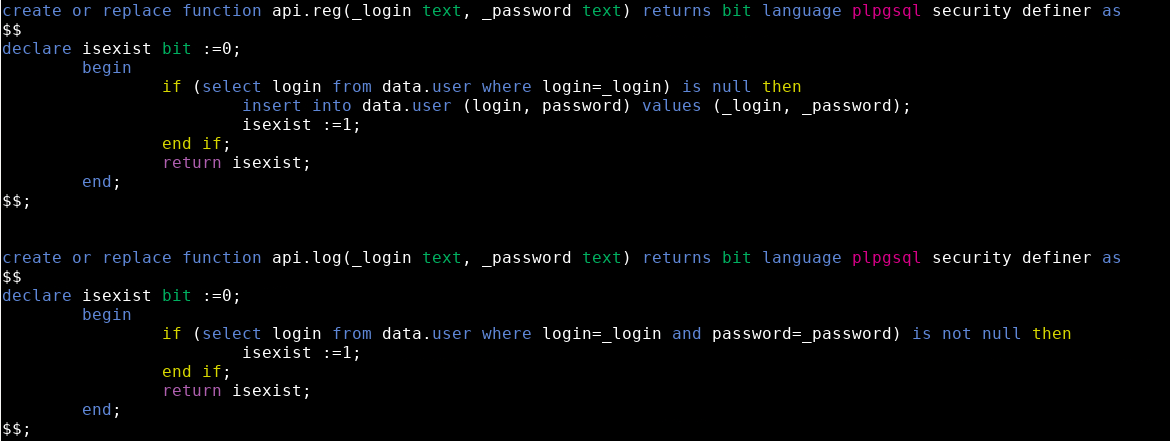


Рисунок 5 - Создание функций api.reg, api.log

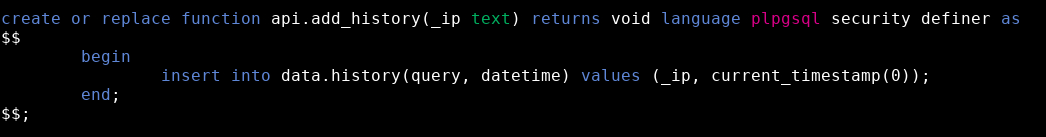
Создание функций api.reg, api.log представлено на рисунке 6 

Рисунок 6 – Создание функции api.add\_history

На рисунке 7 представлена таблица holidays после выполнения функции api.add\_holidays

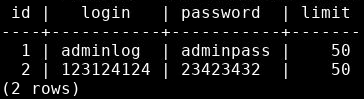


Рисунок 7- Выполнение функции api.add\_holidays

## 2.5. Макет интерфейса приложения

Макет приложения представлен на рисунке 8.

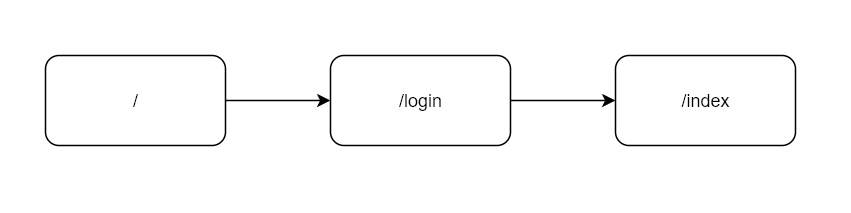


Рисунок 8 - Макет приложения

На рисунке 33 представлена форма Base.



Рисунок 33 - форма Base

Форма Base используется как базовый шаблон для всех страниц приложения.

На рисунке 34 представлена форма index.

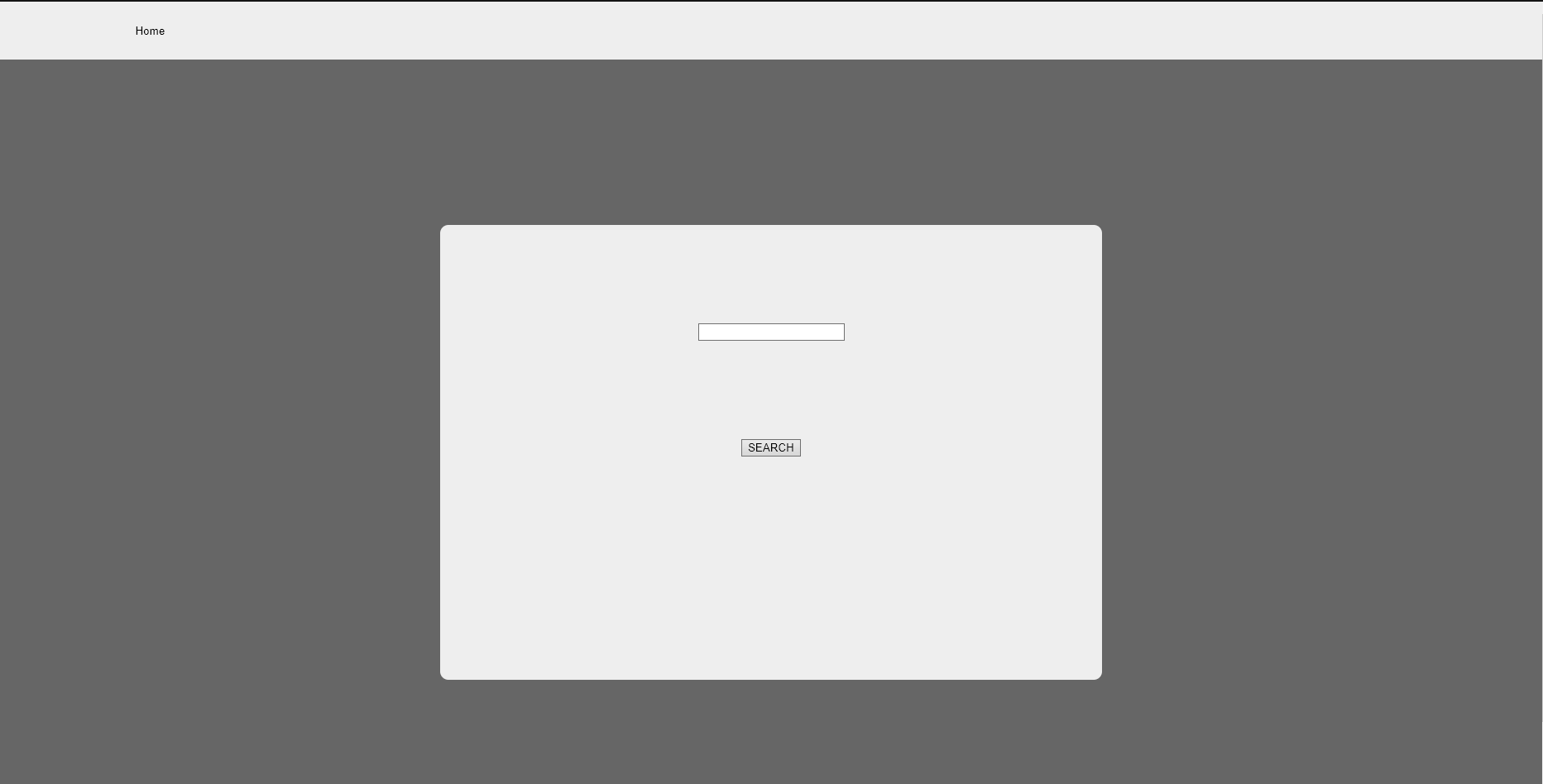


Рисунок 34 - форма index

Кнопка Search используется для вызова функции и возвращения названия страны исходя из ip адреса.

На рисунке 35 представлена форма login.



Рисунок 35 - форма Income information

Кнопка login используется для логина по существующим данным. Кнопка Registration используется для отправления данных для нового пользователя на сервер.

## 2.6. Реализация интерфейса приложения

Код для формы Base представлен на рисунках 37.

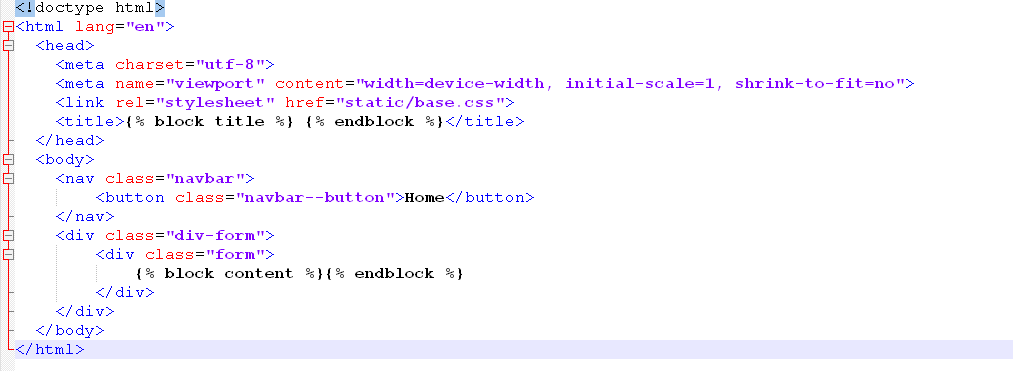


Рисунок 37 - Код для формы 1

Рисунок 38 - Код для формы 1

Код для формы 2 представлен на рисунках 39 и 40.

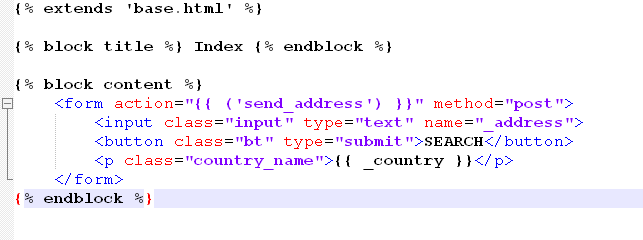


Рисунок 39 - Код для формы 2

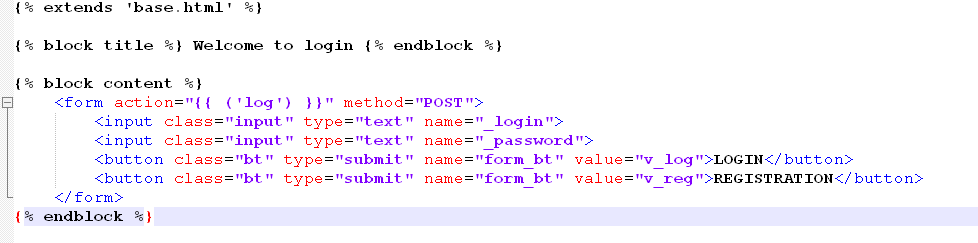


Рисунок 40 - Код для формы 2