ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ

**Отчет по предмету**

**Проектирование и конструирование**

**Программного обеспечения**

по лабораторной работе №7

«Тестирование. Документирование»

Выполнил

Студент ПРО-330Б

Кутлубаев Э.Г.

Проверил

Насыров Р.В.

Уфа 2022

# Цель лабораторной работы

# Ознакомление с видами и методами тестирования. Тестирование. Ознакомление с документами, сопровождающими разработку ПО и их разработка.

# Задание

# 1. Изучить виды и методы тестирования (Приложение 1.А)

# 2. Ознакомиться с тестированием документации (Приложение 1.Б).

# Критически рассмотреть документацию оппонируемой команды (из группы).

# Сформировать перечень замечаний (на соответствие документов – разработке ПО). Обсудить с преподавателем.

# 3. Создать тестовые наборы (данные) для проверки ПО.

# 4. Создать сценарии тестирования (Примеры и рекомендации:

# <https://coderlessons.com/tutorials/kachestvo-programmnogo-obespecheniia/ruchnoetestirovanie/testovyi-stsenarii> ; <https://ru.itpedia.nl/2018/06/01/testcases-voorbeeldenen-best-practices/> ).

# 5. Разработать методику испытаний (примеры и рекомендации <https://www.prjexp.ru/patterns/pattern_methods_of_test.php> ) (Приложение 1.В)

# 6. Разработать программу испытаний (примеры и рекомендации

# <https://www.prj-exp.ru/patterns/pattern_program_of_test.php> ) (Приложение 1.В)

# 7. Провести тестирование.

# 8. Зафиксировать результаты тестирования для отчета (скрины экранов).

# 9. Разработать руководство программиста (Пример. Приложение 2.Б) (структура согласно <https://docs.cntd.ru/document/1200007675> , <https://docs.cntd.ru/document/1200007675> ).

# 10. Разработать руководство пользователя (Пример. Приложение 3.В) (структура документа согласно РД 50-34.698-90 <https://www.prjexp.ru/patterns/pattern_user_guide.php> , стандарт

# <https://www.sibsiu.ru/files/kafedry/avtomatizatsii-i-informatsionnykhsistem/studdocs/%D0%A0%D0%94_50-34.698_90.pdf>)

# Ход выполнения задания

1) Тестирование документации

Тестирование документации - начальная стадия процесса тестирования, которая выступает как система раннего оповещения об ошибках.

Процесс тестирования начинаем с документации и требований.

- мы предположили начало тестирования еще до разработки продукта.

Тестировщик смог указать:

- на логические ошибки в постановке задачи,

- несоответствия в требованиях, а также

- составить чек-лист,

- список проверок по предоставленному требованию.

В процесс тестирования были вовлечены различные специалисты:

- тестировщики,

- проджект-менеджеры,

- бизнес-аналитики,

- разработчики.

Составленная документация обладает следующими свойствами:

∙ требования полные, правильно и в полной мере описывают функцию, которую необходимо было реализовать;

∙ однозначность - одинаковое восприятие требований всеми членами команды, никаких расхождений в трактовке быть не должно;

∙ непротиворечивость: не выявлено противоречивых требований, конфликтующих между собой;

∙ необходимость: требования отражают функциональность, действительно необходимую для пользователя, для удовлетворения пользователей;

∙ осуществимость: насколько прописанных требований возможно реализовать; ∙ тестируемость: возможно проверить все прописанные требования после их реализации, полное требования одинаково воспринимается всеми участниками проекта, ни одна важная деталь не упущена - требование может быть протестировано.

Не были обнаружены следующие часто встречающиеся дефекты документации:

∙ противоречия пунктов/разделов в документе друг другу;

∙ недостаточная детализация требований, их неоднозначное толкование;

∙ нет глоссария, где могли быть указаны все неизвестные термины.

Цель тестирования нашей документации:

∙ проверка корректности требований на предмет полноты требований, их однозначности, осуществимости, непротиворечивости и т.д.;

∙ уменьшения рисков несоответствий реализованного функционала, согласно прописанным требованиям. Наличие таких дефектов в документации приводит к значительному увеличению расходов и времени, затраченном на исправление допущенных ошибок;

∙ предотвратить допущение ошибок разработчиком при написании кода;

∙ уменьшение рисков передачи в эксплуатацию программного продукта с некачественной документацией.

Перспективы использования тестирования документации:

∙ повышение качества реализаций;

∙ развивает аналитические навыки тестировщиков;

∙ дифференцирует нагрузку на тестировщиков (выявление потенциально возможных ошибок может существенно уменьшить время на тестирование конкретно этой функциональности, следовательно, это время можно потратить на проведение, например, исследовательского тестирования).

Тестирование документации в нашем случае наиболее актуально, т.к.:

∙ заказчик активно участвует в разработке проекта, он принимает каждый новый релиз;

∙ заказчик имеет доступ к документации и может контролировать ее актуальное состояние.

Помимо уменьшения рисков, устранения несоответствий, сформированное нами тестирование документации решает важные вопросы, касающиеся бизнес-целей проекта:

∙ сокращение затрат на техническую поддержку (за счет быстрого нахождения ответов в документации);

∙ чем подробнее описана функциональность, тем проще будет ее протестировать в полном объеме;

∙ сокращение затрат на разработку новой функциональности за счет уменьшения расходов, в случае некачественного описания требований в документации. Какая документация подвержена тестированию:

∙ продуктная документация - план проекта, требования к ПП, функциональные спецификации, архитектура и дизайн, тест-кейсы, технические спецификации;

∙ проектная документация [понятие более широкое] - включает в себя продуктную документацию, а также пользовательскую и сопроводительную документацию, маркетинговую документацию.

Во время разработки проекта, все, что касается разрабатываемого продукта, будь то наброски маркером на доске, переписка в скайпе, почтовая переписка - все является своего рода документацией. И все подвержено тестированию. Перечитываются письма, которые отправляются заказчику, чтобы не допустить ошибок и не упустить важное.

Тестирование документации и требований относится к разряду нефункционального тестирования.

Тестирование документации проводилось по следующим специальным техникам для тестирования документации и требований:

1. review требований.

∙ беглый просмотр - показ своей работы коллегам с целью получения обратной связи, вопросов и замечаний. Все отзывы и замечания помогли улучшить работу.

∙ технический просмотр - вычитка документа группой специалистов, представляющих различные области. Документ/требования являются быть качественными, т.к. ни у одного из специалистов нет замечаний.

∙ формальная инспекция - редко используемая техника (например, при получении проекта, созданием которого занималась другая компания, на сопровождение и доработку), которая представляет собой систематизированный подход к анализу документации.

2. вопросы - одна из наиболее простых и эффективных техник выявления требований. Если что-то в документах оставалось непонятным – мы задавали вопросы. Для получения ответов на задаваемые вопросы, обращались к менеджеру, более опытному специалисту, который ранее получил соответствующую информацию от заказчика.

3. тест-кейсы - требование проверяемо, существуют способы проверки корректности реализованного требования. Чек-лист продуман тщательно, а значит вероятнее определение проверяемости требований. Прежде, чем записать возможные тест-кейсы, мы убедились в том, что понимаем требование. Для хорошего понимания конкретного требования проанализировали другие требования, которые были связаны. Когда все требования были хорошо сформулированы и протестированы, мы продолжили использование этой техники, совмещая разработку тест-кейсов с дополнительным тестированием.

4. исследование поведения системы: тестировщик смоделировал процесс работы системой, созданной по тестируемым требованиям, и нашел неоднозначные варианты поведения системы.

**Тестовые наборы для проверки ПО**

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 1 - Пустая

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 2 - Пустые ячейки

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3 - Таблица с отсутсвующим столбцом

**Сценарии тестирования**

Backend:

* Проверьте поведение системы при вводе тестового набора без значений;
* Проверьте поведение системы при вводе тестового набор, содержащего пустые ячейки;
* Проверьте поведение системы при вводе тестового набора без названия столбцов;
* Проверьте поведение системы при вводе пустого тестового набора;
* Проверьте поведение системы при вводе тестового набора без одного столбца;
* Проверьте поведение системы при вводе тестового набора с неподдерживаемым расширением;

Frontend:

* Проверьте поведение системы при обращении к странице arenda.html, при отсутствии данных в БД.

**Объект испытаний**

Полное наименование – система автоматической обработки данных

Обозначение - система

**Цель испытаний**

Целью проведения испытаний является:

- проверка взаимодействия подсистем Системы;

- проверка работоспособности Системы;

- проверка соответствия Системы требованиям, приведенным в документе «Техническое задание»;

- проверка готовности Системы к проведению опытной эксплуатации или приемочных испытаний на территории Заказчика.

**Требования к системе**

* 1. Когда сервис-обработчик обнаруживает файл, он должен узнать его расширение в течение секунды.
  2. Когда сервис-обработчик определяет расширение файла, он должен запустить соответствующий фабричный метод, если расширение допустимо.
  3. Если сервис-обработчик не может считать файл, он должен вернуться в состояние ожидания файла в течение двух секунд.
  4. Когда сервис-обработчик открывает файл, он должен запустить методы обработки в течение 2 секунд.
  5. Когда сервис-обработчик обработал данные, он должен установить соединение с SQL сервером в течение минуты.
  6. Если сервис-обработчик не получил ответа от SQL сервера, то он должен вернуться в состояние ожидания файла в течение двух секунд.
  7. Если сервис-обработчик получил ответ от SQL сервера, он должен загрузить обработанные данные в БД в течение трех секунд.
  8. Если сервис-обработчик загрузил данные в БД он должен вернуться в режим ожидания файла в течение трех секунд.
  9. Если сервис-обработчик получил сигнал на прерывание, он должен завершить свою работу в течение трех секунд.

**Требования к программной документации**

К моменту тестирования должны быть представлены следующие документы:

1. Техническое задание.
2. Программа и методы тестирования.

**Средства и порядок испытаний**

Для тестирования необходимы:

1. Python 3
2. Набор браузеров для тестирования кроссплатформенности.

**Методы испытаний**

**Испытание 1** – Функциональное тестирование. Проверка работоспособности при различных входных данных.

1. Запустить систему.
2. Загрузить тестируемый набор данных.
3. Отметить результат.

**Испытание 2** – Usability тестирование

1. Проверить все ли кнопки понятны в использовании.
2. Проверить отсутствие грамматических/орфографических ошибок.
3. Проверить понятна ли структура веб-приложения.
4. Проверить работу переходов при нажатии на элементы навигационного меню.

**Испытание 3** – Тестирование совместимости

1. Запустить сайт на различных браузерах.
2. Результаты тестирования

**Испытание 1**

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4 - Результат обраотки файла без значений. Файл обработан, БД обновилась, на сайте нет характеристик о вертолетах.

Изображение выглядит как текст

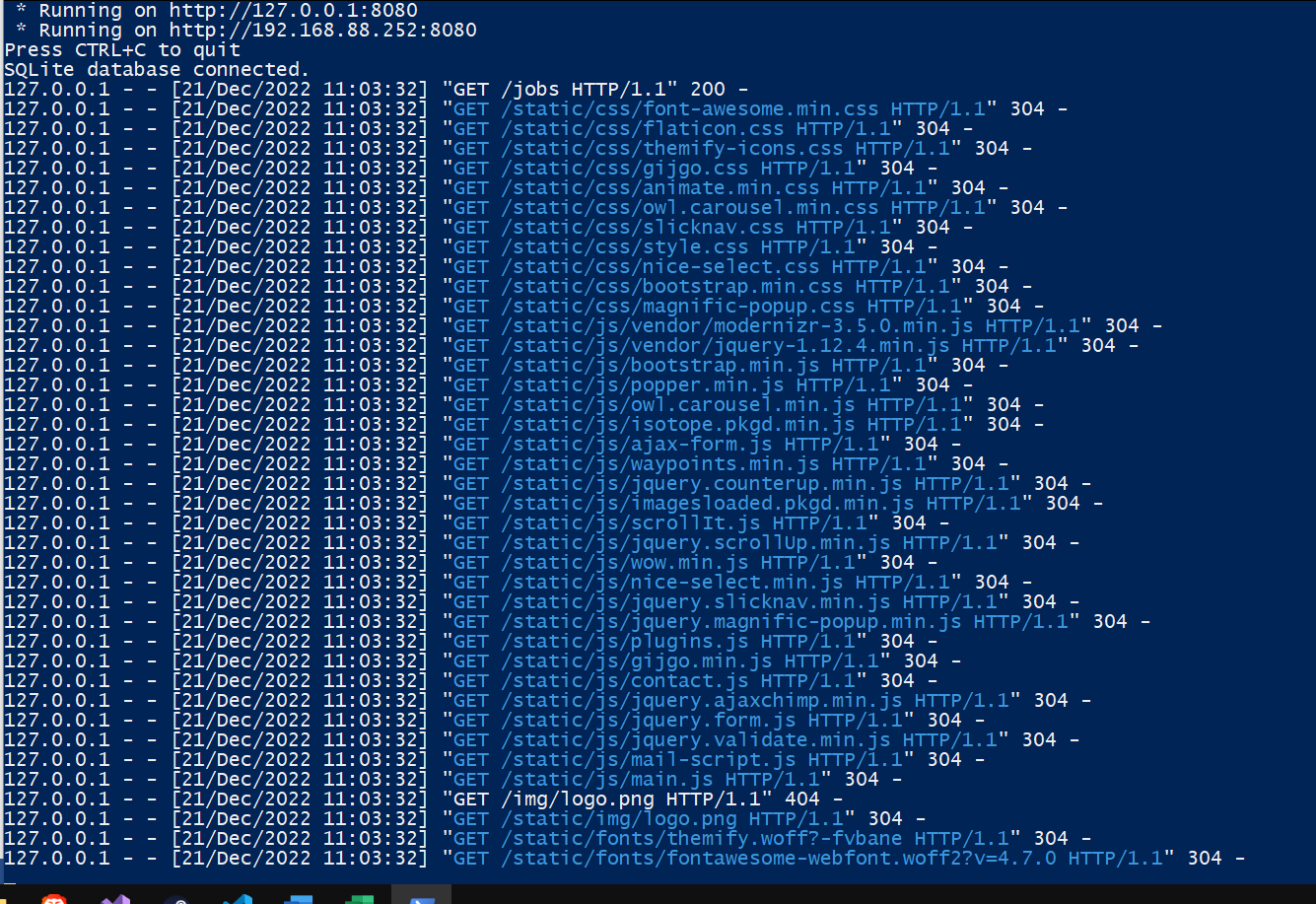
Автоматически созданное описание

Рисунок 5 - Результат обработки таблицы с некоторым пустыми значениями. Обработчик работает. БД не обновлялась.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6– Результат обработки c пустыми таблицами. БД не обновлялась.



Руководство программиста.

См. Приложение 1.

1. Руководство пользователя.

См. Приложения 2.

# Приложение 1

### Руководство программиста

### Аннотация

В данном программном документе приведено руководство программиста по настройке и использованию сервиса по обработке статистических данных о суициде и предоставлению их пользователю.

В разделе «Назначение и условия применения программы» указаны назначение и функции, выполняемые программой, условия, необходимые для выполнения программы (объем оперативной памяти, требования к составу и параметрам периферийных устройств, требования к программному обеспечению и т. п.).

В разделе «Характеристика программы» приведено описание основных характеристик и особенностей программы.

В разделе «Входные и выходные данные» приведено описание организации, используемой входной и выходной информации.

### Назначение и условие применения программы

### Назначение программы

Программный продукт предназначен для обработки статистических данных о суициде и предоставлению их пользователю.

### Функции, выполняемые программой

В программе реализованы следующие функции:

* Автоматическая загрузка файла с данными из заданной папки
* Обработка данных.
* Отправка полученных после обработки данных в базу данных.
* Предоставление содержания БД пользователю на сайте.

### Условия, необходимые для выполнения программы

Для обеспечения работоспособности допустимыми являются характеристики:

* операционная система – Microsoft Windows 7/8/10;
* тип системы – 32 или 64 бит;
* минимальный процессор – 2 ГГц;
* минимальный объем оперативной памяти – 2 ГБ;
* жесткий диск объемом не менее 2 Гб

### Программные условия для запуска программы

Для работы программы требуются

* python 3.7.6+;
* библиотеки для Python: – threading, time, pandas, sqlite, pymysql, flask.

### Характеристика программы

### Режим работы программы

Программа представляет собой приложения для персонального компьютера, которая должна работать в режиме реального времени при соблюдении корректности входных данных и спецификации программного и аппаратного обеспечения.

### Функционирование программы после сбоев

Функционирование программы после сбоев возможно после перезапуска программы

### Обращение к программе

Запуск программы осуществляется через командную строку, где сначала необходимо выбрать путь в папку программы, а затем выбрать главный файл командной «python main.py».

### Входные и выходные данные

### Формат, характер и организация входных данных

В качестве входных данных могут использоваться csv или txt файлы со следующими стоблцами: id, first\_name, last\_name, country, city, email.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 17 - Пример входных данных

### Формат, характер и организация выходных данных

Выходным данными являются результаты обработки входных данных, представленные на веб-сайте.

### Язык программирования и инструментальные средства разработки

В качестве основного языка программирования был выбран Python поскольку он обеспечивает довольно короткие сроки написания программ, также имеется большое количество библиотек, реализующих подключение к БД, запуск веб-приложения и работу с табличными данными. Программа разрабатывалась в среде разработки VSCode.

### Структура программы

Ниже представлена структура программы.

Изображение выглядит как текст, табло, снимок экрана

Автоматически созданное описание

Рисунок 7 – структура программы

База данных имеет следующую структуру:

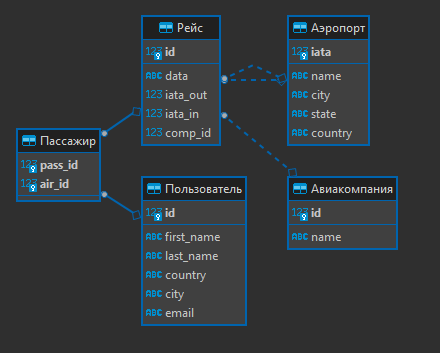


Рисунок 8 – структура базы данных

Таблица Аэропорт содержит код, название, город, страну и область. Рейс имеет внешние ключи с Аэропортом и Авиакомпанией. Авиакомпания содержит уникальный код и название. Пользователь, этот клиент кто зарегистрировался в системе и может купить билет.

Приложение 2

### Руководство пользователя

### Аннотация

В данном программном документе приведено руководство пользователя по применению и эксплуатации сервиса по обработке данных о полетах на вертолетах и предоставлению их пользователю.

В разделе «Назначение программы» указаны сведения о назначении программы и информация, достаточная для понимания функций программы и ее эксплуатации.

В разделе «Условия выполнения программы» указаны условия, необходимые для выполнения программы (минимальный состав аппаратных и программных средств и т. п.).

В разделе «Выполнение программы» указана последовательность действий пользователя, обеспечивающих запуск, выполнение и завершение программы, а также приведено описание всех функций. Назначение программы

### Функциональное назначение программы

Программный продукт предназначен для обработки заказов на полеты на вертолетах и предоставление информации о них пользователю.

### Эксплуатационное назначение программы

Ограничений на эксплуатацию программы не вводится

### Состав функций

В программе реализованы следующие функции:

* Автоматическая загрузка файла с данными из заданной папки
* Обработка данных.
* Отправка полученных после обработки данных в базу данных.
* Предоставление содержания БД пользователю на сайте.

### Условия выполнения программы

### Минимальный состав аппаратных средств

Для обеспечения работоспособности допустимыми являются характеристики:

* операционная система – Microsoft Windows 7/8/10;
* тип системы – 32 или 64 бит;
* минимальный процессор – 2 ГГц;
* минимальный объем оперативной памяти – 2 ГБ;
* жесткий диск объемом не менее 2 Гб

### Минимальный состав программных средств

Для работы программы требуются

* python 3.7.6+;
* библиотеки для Python: – threading, time, pandas, sqlite, pymysql, flask.

### Требования к пользователю

Программа рассчитана на пользователя, имеющего базовые знания о работе в ОС Windows

### Выполнение программы

Запуск программы осуществляется через командную строку, где сначала

необходимо выбрать путь в папку программы, а затем выбрать главный файл

командной «python main.py» (рис. 18).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9 – Запуск программы через командую строку.

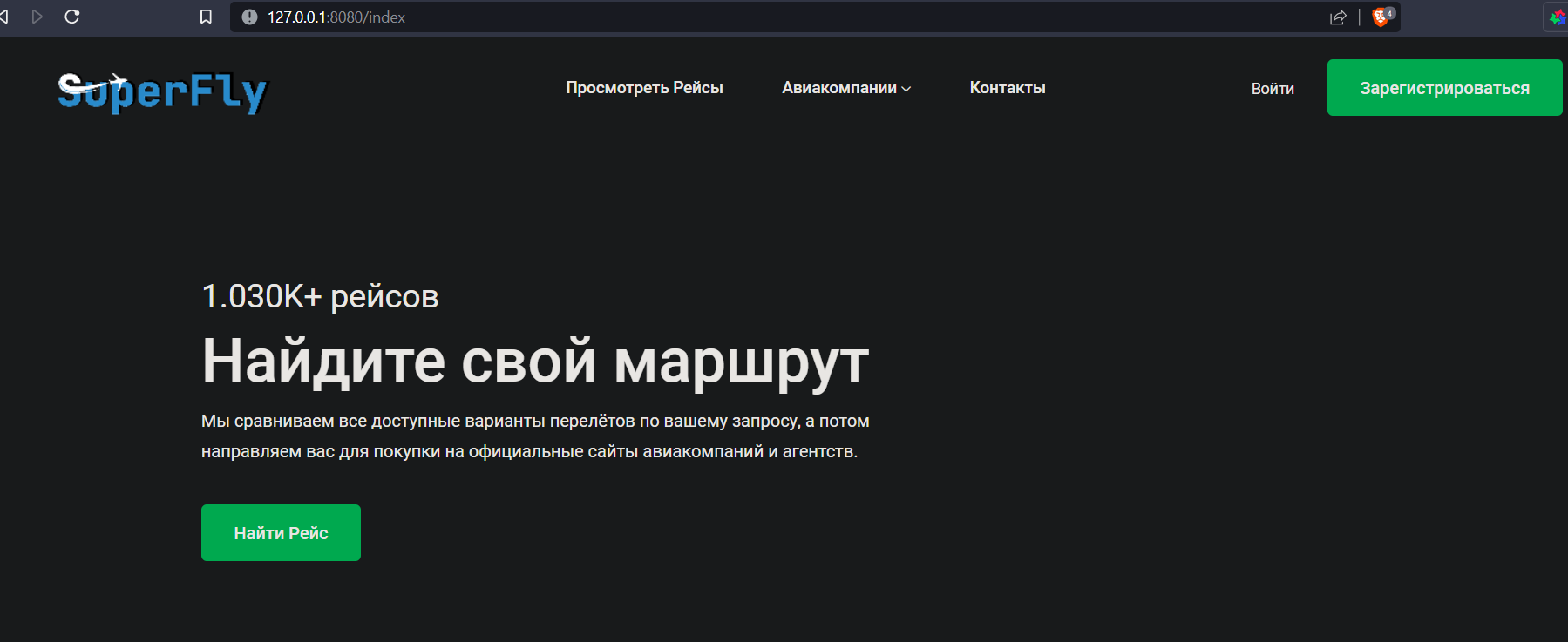


Рисунок 10 – Отображение при открытии адрес

**GitHub**

1. <https://github.com/EmilK102/proectirovanie>