

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский Авиационный Институт»  
(Национальный Исследовательский Университет)

Институт 8 «Информационные технологии и прикладная математика»  
Кафедра 806 «Вычислительная математика и программирование»

**Курсовой проект**  
**по курсу «Базы данных»**  
**«Система регистрации пассажиров аэропорта»**

**Выполнил:** Даутов Т. Б.

**Группа:** М8О-306Б-22

**Преподаватель:** Малахов А. В.

Москва, 2024

## **Содержание**

|   |          |
|---|----------|
| <b>Постановка задачи и формальные требования.....</b> | <b>3</b> |
| <b>Схема базы данных.....</b>                         | <b>4</b> |
| <b>Использованные технологии и инструменты.....</b>   | <b>5</b> |
| <b>Описание приложения и функционала.....</b>         | <b>6</b> |
| <b>Выводы.....</b>                                    | <b>9</b> |

## **Постановка задачи и формальные требования**

1. При реализации курсового проекта допускается только использование СУБД PostgreSQL.
2. Необходимо выбрать предметную область для создания базы данных. Выбранная предметная область должна быть уникальной для всего потока, а не только в рамках учебной группы.
3. Необходимо описать модели предметной области и уровня инфраструктуры и их назначение в рамках реализуемого проекта (минимальное количество моделей предметной области и уровня инфраструктуры - 5). Также необходимо выполнить проектирование логической структуры базы данных. Все таблицы, связанные с описанными моделями предметной области, должны находиться в 3NF или выше. База данных должна иметь минимум 7 таблиц.
4. Необходимо разработать клиентское приложение (B2C) для доступа к информации из базы данных. Реализованное приложение должно быть понятно в плане использования (“приложение для домохозяек”). Выбор языков программирования и технологий разработки клиентского приложения произволен: C/C++, python, perl, ruby, JavaScript, php, swift, Java и др.
5. Необходимо организовать различные роли пользователей и права доступа к данным (например: администратор, редактор, рядовой пользователь). Клиентское приложение, взаимодействующее с базой данных, должно предоставлять функционал для авторизации пользователя по логину и паролю (хранение непосредственно пароля в базе данных запрещено).
6. Необходимо реализовать возможность создания администратором архивных копий базы данных и восстановления данных из клиентского приложения.

7. При разработке функционала базы данных следует организовать логику обработки данных не на стороне клиента, а на стороне серверного приложения (API), при этом клиентские приложения служат только для представления данных, выполнения запросов к данным и тривиальной обработки данных. Запросы к данным из базы данных должны выполняться асинхронно как для клиентского, так и для серверного приложения.
8. На стороне базы данных необходимо определить представления, триггеры, функции и хранимые процедуры, причем все эти объекты должны быть осмыслены, а их использование оправдано.
9. При демонстрации Вашего проекта необходимо уметь демонстрировать реализованный функционал уровня БД, уровня серверного приложения, уровня клиентского приложения; уметь демонстрировать подключение к базе данных, основные режимы работы с данными (просмотр, редактирование, обновление и т. д.). Также необходимо подготовить скрипт SQL для инициализации структуры базы данных.
10. Необходимо реализовать корректную обработку различного рода ошибок, которые могут возникать при работе с базой данных.

### **Схема базы данных**

База данных системы состоит из восьми таблиц. Содержание таблиц и связи между ними представлены на рисунке ниже.

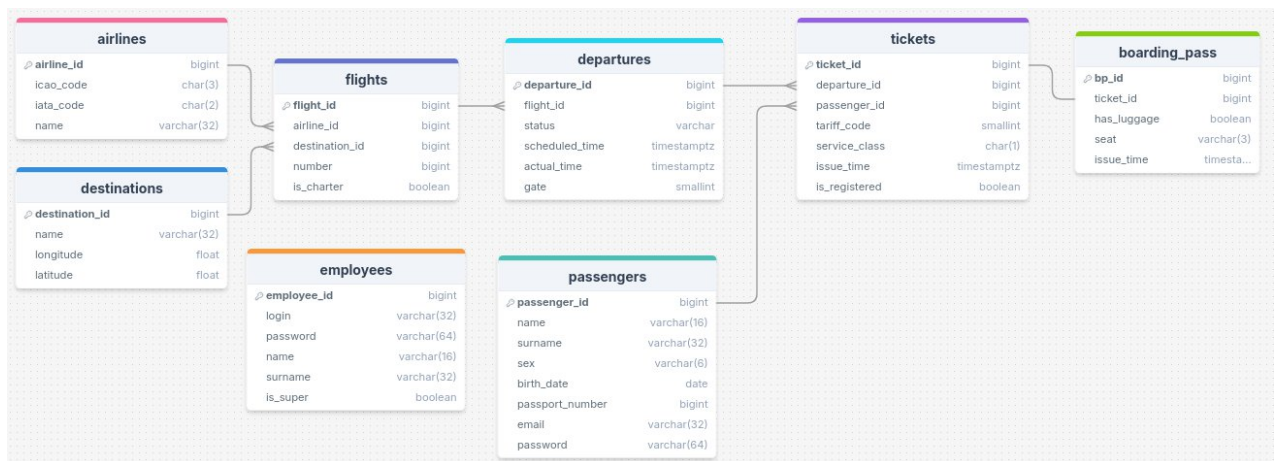


Рис. 1 – Диаграмма БД

Список таблиц и их назначение:

- airlines (Авиакомпания): содержит информацию об авиакомпаниях.
- destinations (Направление): информация о направлениях.
- flights (Рейс): содержит информацию о рейсах.
- departures (Отправление): хранит записи о запланированных отправлениях.
- tickets (Билет): хранит информацию об оформленных билетах.
- boarding\_pass (Посадочный талон): содержит информацию о выданных посадочных талонах.
- employees (Сотрудник): содержит записи о сотрудниках аэропорта.
- passengers (Пассажир): содержит информацию о пассажирах.

## Использованные технологии и инструменты

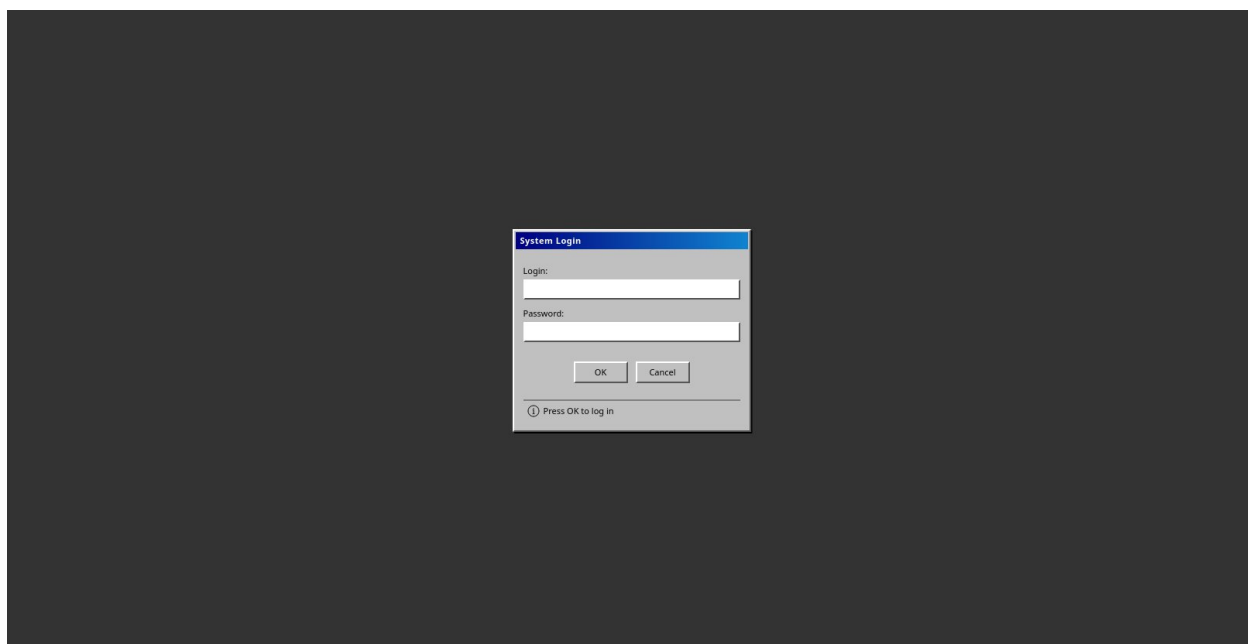
При выполнении курсовой работы были использованы следующие инструменты:

- СУБД **PostgreSQL** (Популярная и надёжная реляционная система управления базами данных, обладающая широким функционалом)
- **FastAPI** (современный Python фреймворк для построения API и приложений, основанный на Starlette)
- **asyncpg** (Python модуль, предоставляющий асинхронный драйвер для взаимодействия с СУБД PostgreSQL)
- **passlib** (Python модуль, содержащий функционал для хэширования паролей)
- **Docker** (Популярная система контейнеризации)

- Frontend часть приложения реализована с помощью стандартных средств **HTML, CSS, JavaScript** и шаблонизатора **Jinja2**.

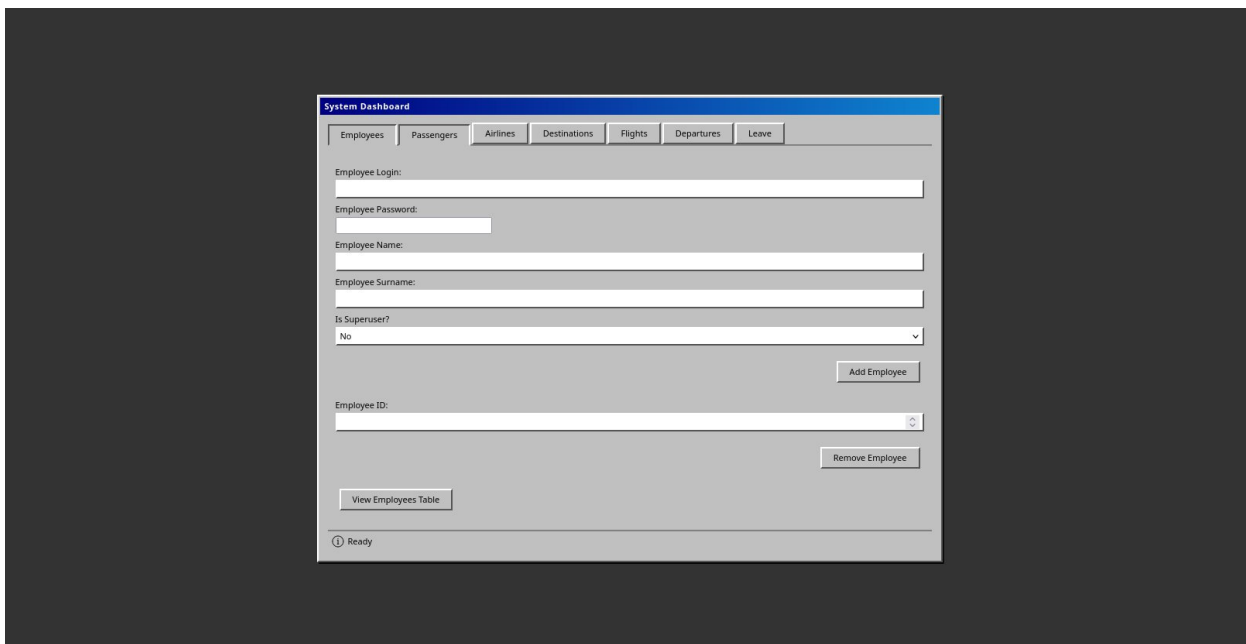
## Описание приложения и функционала

Приложение имеет механизм отдельной авторизации для сотрудников аэропорта (эндпоинт «/emplogin»). В ней сотрудники могут авторизоваться по логину и паролю и получить доступ к своему рабочему месту (панели, эндпоинт «/workplace») в соответствии с их статусом.



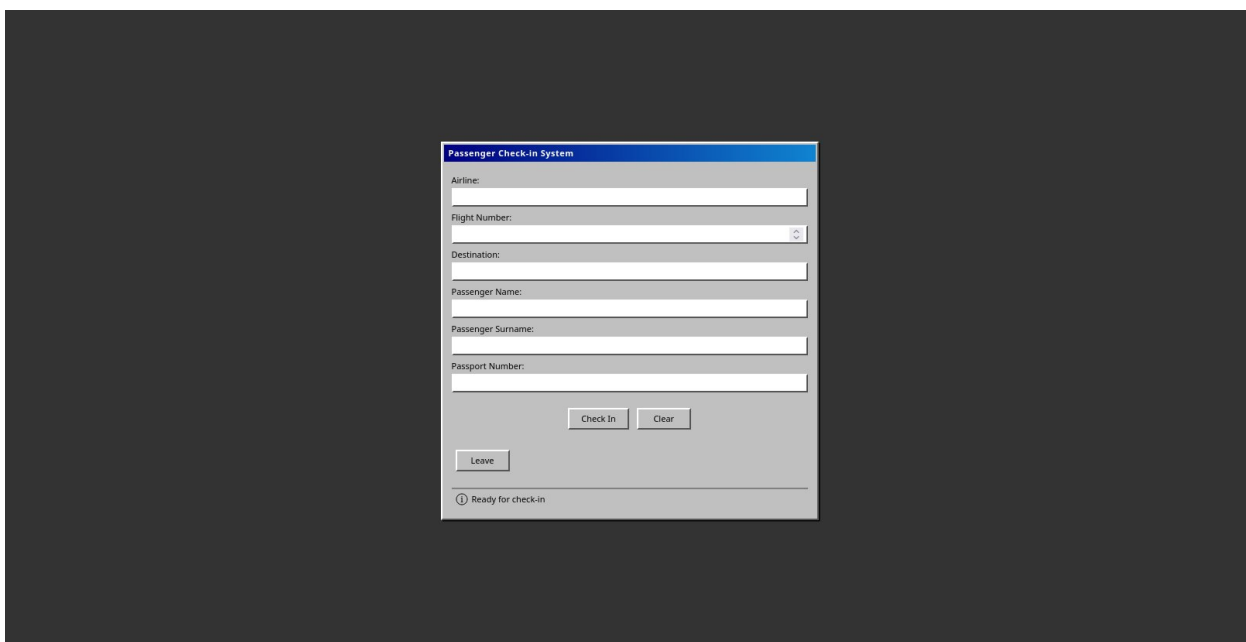
*Рис. 2 – Страница входа для сотрудников*

Панель администратора имеет вкладочный интерфейс, в каждой вкладке администратор может просматривать, вносить и удалять информацию, связанную с соответствующей таблицей (сущностью).



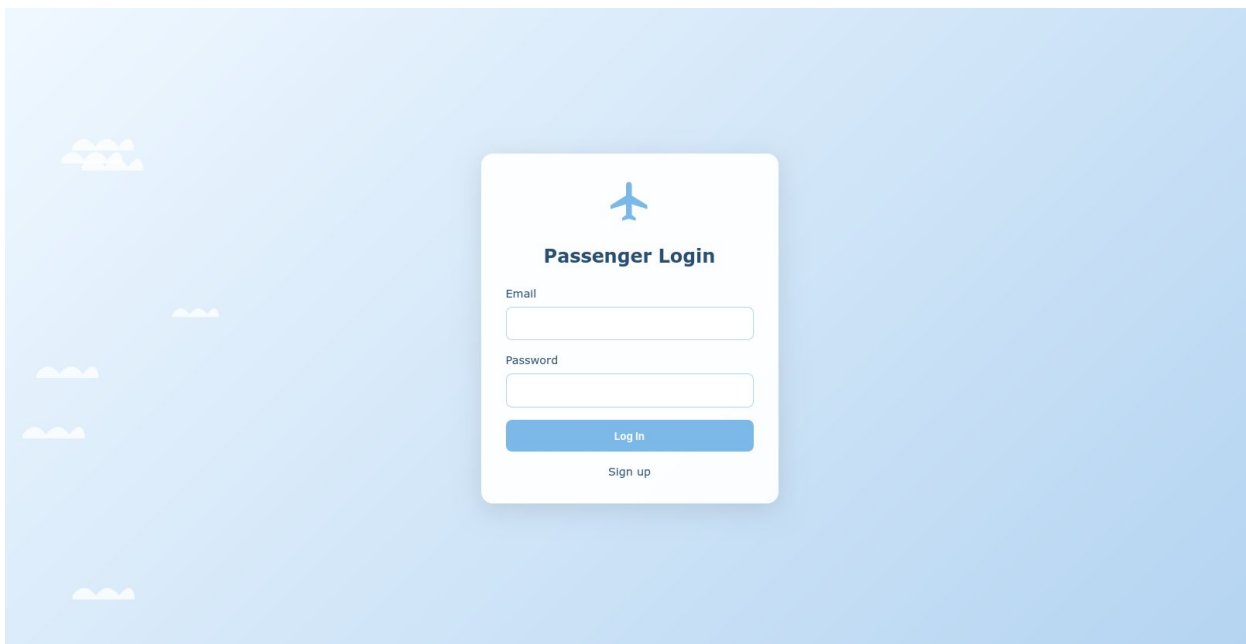
*Рис. 3 – Панель управления администратора*

Панель сотрудника регистрации содержит форму для внесения регистрационных данных пассажира. После заполнения формы следует нажать кнопку «Check In», после чего система проверит данные и выдаст посадочный талон.



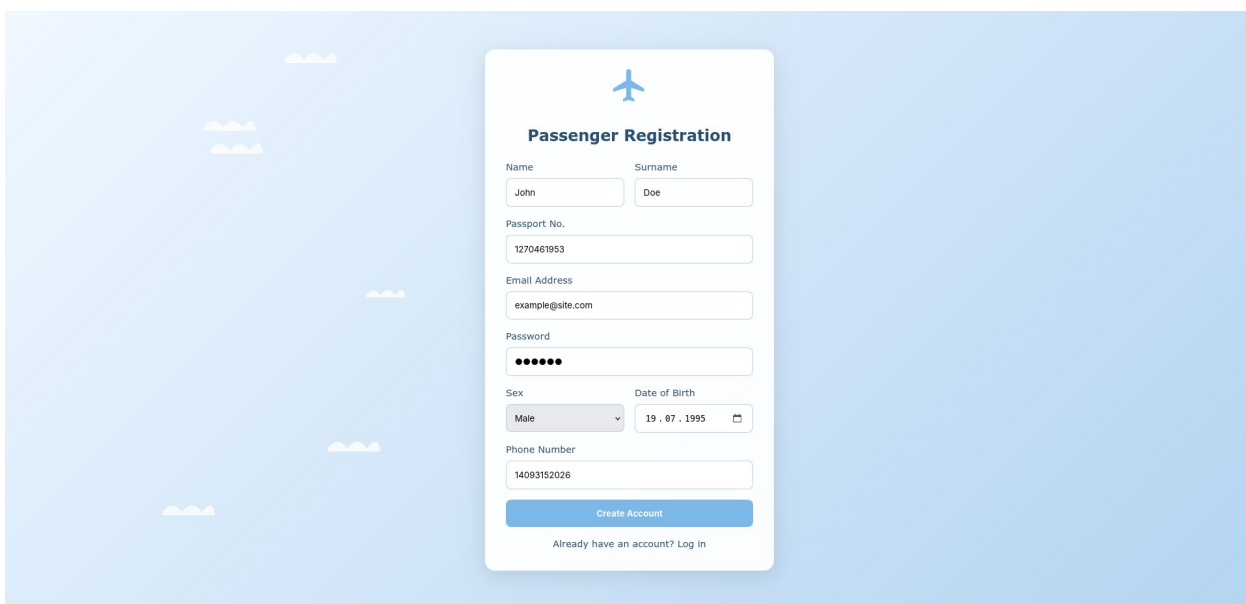
*Рис. 4 – Панель сотрудника регистрации*

Для пользователей (пассажиров) реализована форма входа по электронной почте и паролю (эндпоинт «/login»).



*Рис. 5 – Страница входа пользователей (пассажиров)*

Страница регистрации пользователя (эндпоинт «/signup») содержит соответствующую форму.



*Рис. 6 – Страница регистрации пользователей (пассажиров)*

Страница онлайн-расписания (эндпоинт «/timetable») содержит фильтр по датам, а также таблицу найденных по запросу рейсов.



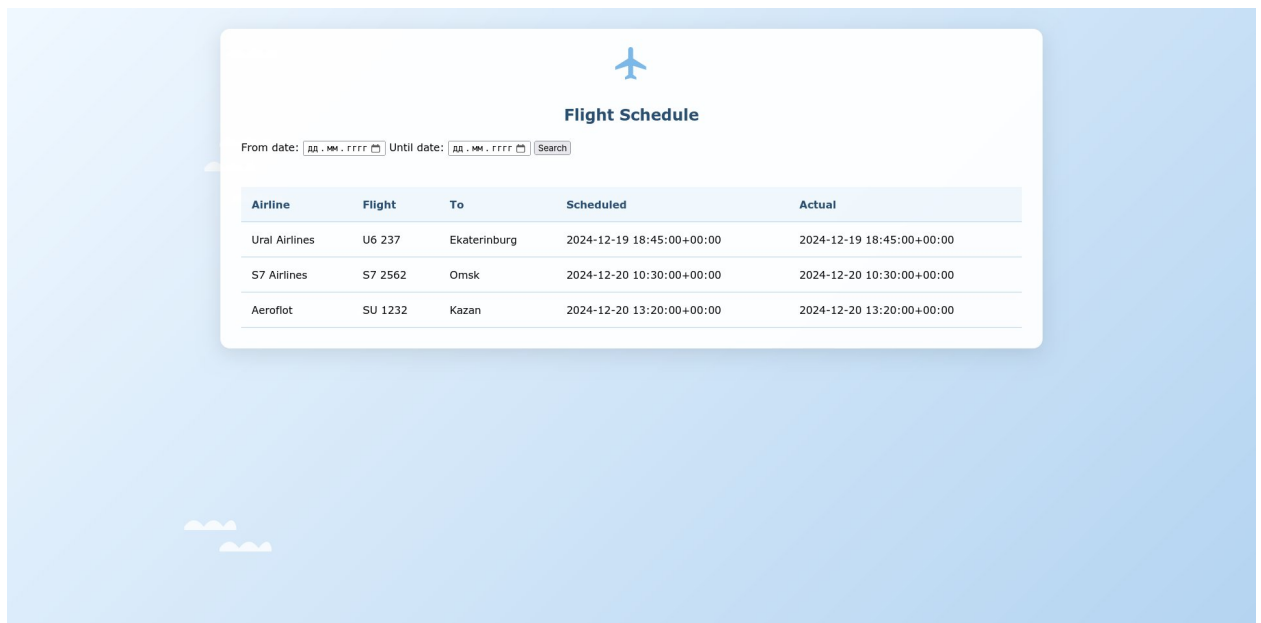


Рис. 7 – Страница онлайн-расписания вылетов

Помимо прочего, есть страница для бронирования билетов на предстоящие отправления (эндпоинт «/shop») с возможностью выбора класса обслуживания.

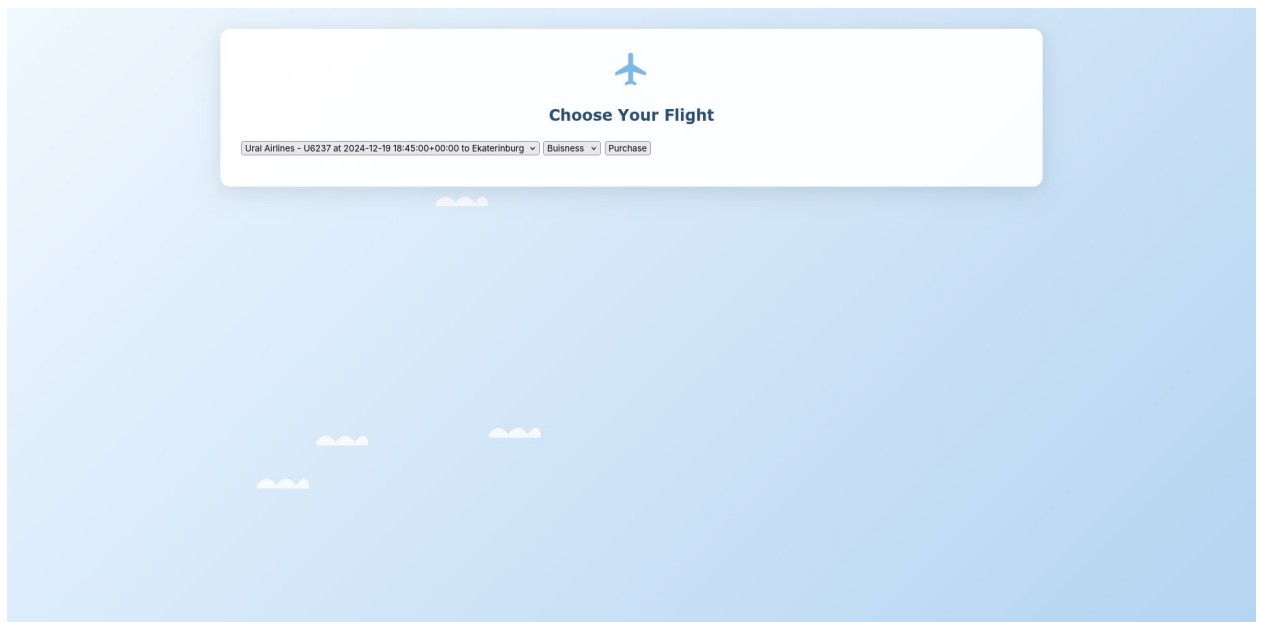


Рис. 8 – Страница оформления билетов

## Исходный код

Ссылка на GitHub репозиторий: [https://github.com/s1tkeyz/db\\_project](https://github.com/s1tkeyz/db_project)



Рис. 9 – QR код на репозиторий

## Выводы

В ходе выполнения курсового проекта мной было построено Web-приложение на языке программирования Python, взаимодействующее с СУБД PostgreSQL. Я приобрёл новые навыки и умения, подкрепил теоретические знания практикой. Была реализована составленная структура БД, а также триггеры, функции и представления. Данный опыт, несомненно, пригодится в дальнейшей работе в этом направлении.