**2 слайд**

**Основные цели и задачи проекта**

Цель: создать приложение, применяя язык программирования Python, библиотеку PyQt5 и базу данных SQLite.

Задачи:

* Разработать идею
* Продумать архитектуру проекта
* Заполнить базу данных
* Написать программу
* Разработать внешний дизайн приложения

Идея: Приложение для покупки билетов в театр.

Описание: Мой проект представляет собой демонстрационную версию приложения по покупки билетов. В нем рассматривается только один театр, но приложение создано так, что для добавления театров не требуется больших усилий.

**3 слайд**

**Архитектура проекта и техническое описание**

Программа состоит из нескольких файлов. В файле main.py находится основная часть программы, где описаны классы форм. Accepting.py, play.py, repertoire.py, start\_page.py, theatre.py – файлы с классами дизайна форм. Файл import\_data.py используется для работы с базой данных.

Рассмотрим некоторые фрагменты программы более подробно.

**4 - 6 слайд**

Для получения списка спектаклей мы создаем экземпляр класса Import\_export()

importer = Import\_export()

В self.result мы импортируем все спектакли из таблицы 'repertoire'

В self.d в качестве ключа мы сохраняем название спектакля, а в качестве значения все остальные необходимые данные.

На форме с репертуаром доступны два вида поиска: по имени и по дате. Чтобы сделать поиск в реальном времени, необходимо создать два экземпляра класса QTimer. За одинаковые промежутки времени они обновляют listWidget.

**7 слайд**

Чтобы заполнить форму отдельного представления, мы получаем список всех спектаклей по датам и среди них ищем те, в которых встречается название выбранного.

**8 слайд**

Объявляем экземпляр класса Play\_form() , где в качестве аргументов передаем необходимые данные, возможные даты и базу данных. Далее заполняем форму.

**9 слайд**

Чтобы иметь возможность обращаться к каждой кнопке, нам необходимо добавить их в список. Из-за большого количества мест этот фрагмент был добавлен в файл с классом дизайна формы. Далее мы проходимся по всем кнопкам и присваиваем подходящий для нее цвет. Если в базе данных у данного места стоит нулевая стоимость, то соответствующая кнопка становится недоступной.

**10 слайд**

В self.chosen у нас сохранены выбранные кнопки. Мы проходимся по этому списку и определяем местоположение кнопки в зале по ее имени.

**11 слайд**

В данном фрагменте выполняется отправка email. Используется библиотека smpt и сервер Yandex для автоматической отправки email. При ошибке отправки приложение сообщает пользователю об этом и отменяет покупку.

**12 слайд**

База данных заполнялась при помощи парсеров parser\_placex.py, parser\_repertoire.py и update\_data.py, которые получали данные с сайта театра. Парсинг выполняется при помощи библиотек bs4 и Selenium.

**13 слайд**

Таким образом, это приложение для покупки билетов, которое позволяет без больших усилий подключать к данному сервису новые театры путем добавления новых баз данных, заполненных по тому же принципу, что и стандартная.

Чтобы сделать приложение максимально функциональным, необходимо загрузить базу данных на удаленный сервер и добавить реальную покупку.