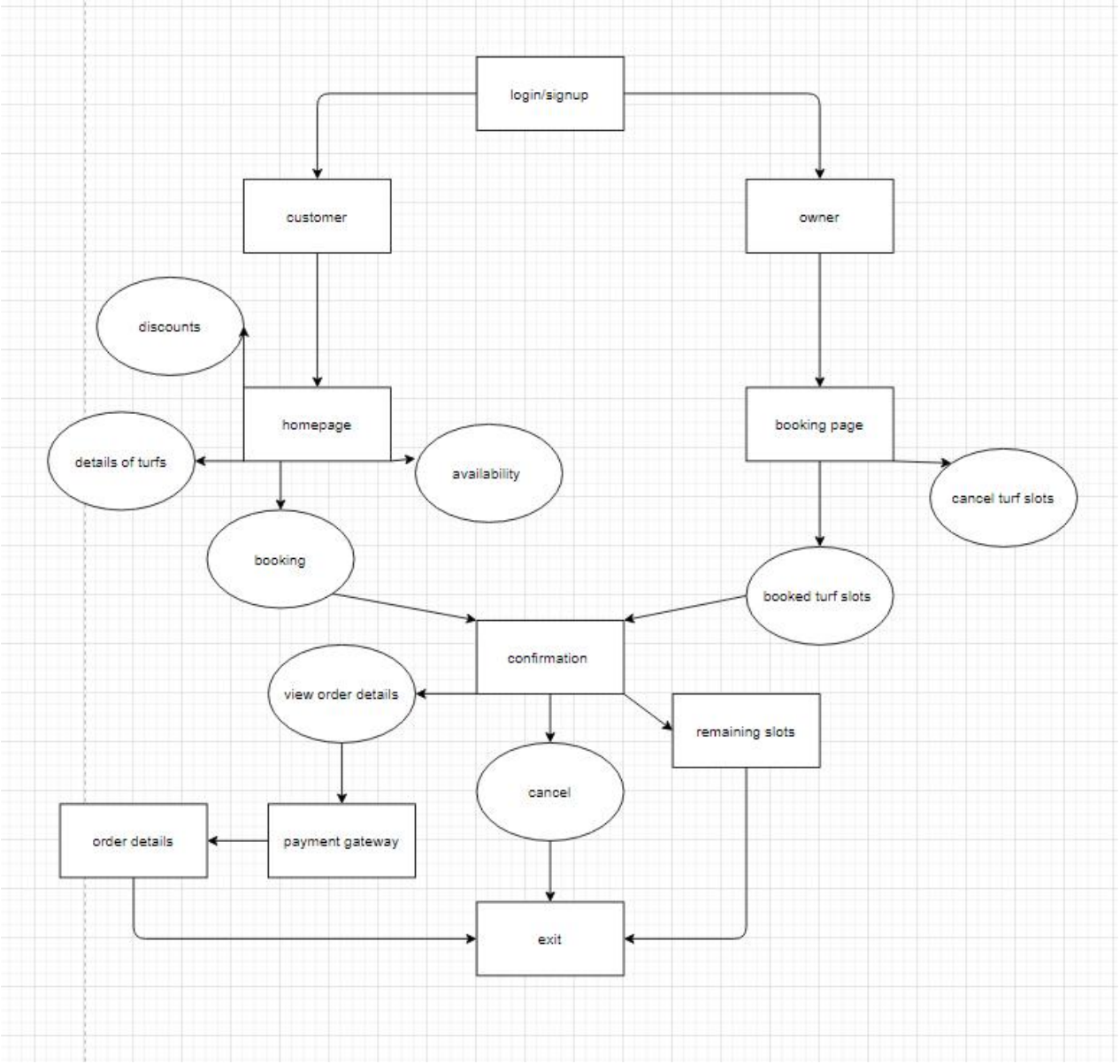
Sports-Booking-System

Introduction

It is a web application that simplifies the process of booking sports venues and events. Built using Django on the backend, it offers users a seamless way to explore, filter, and book facilities. Administrators can efficiently manage schedules, venues, and user bookings. This system aims to streamline the booking process, prevent scheduling conflicts, and enhance the overall user experience.

System Design



Technology Used

1. Frontend :

a. HTML: HTML is the standard markup language for Web pages.We used it to design our web pages.

b. CSS: CSS is the language we used to style our pages.

c. JavaScript: We used JavaScript to program the behavior of web pages

d. Bootstrap:Bootstrap was used to improve the design of the web pages and make them responsive.

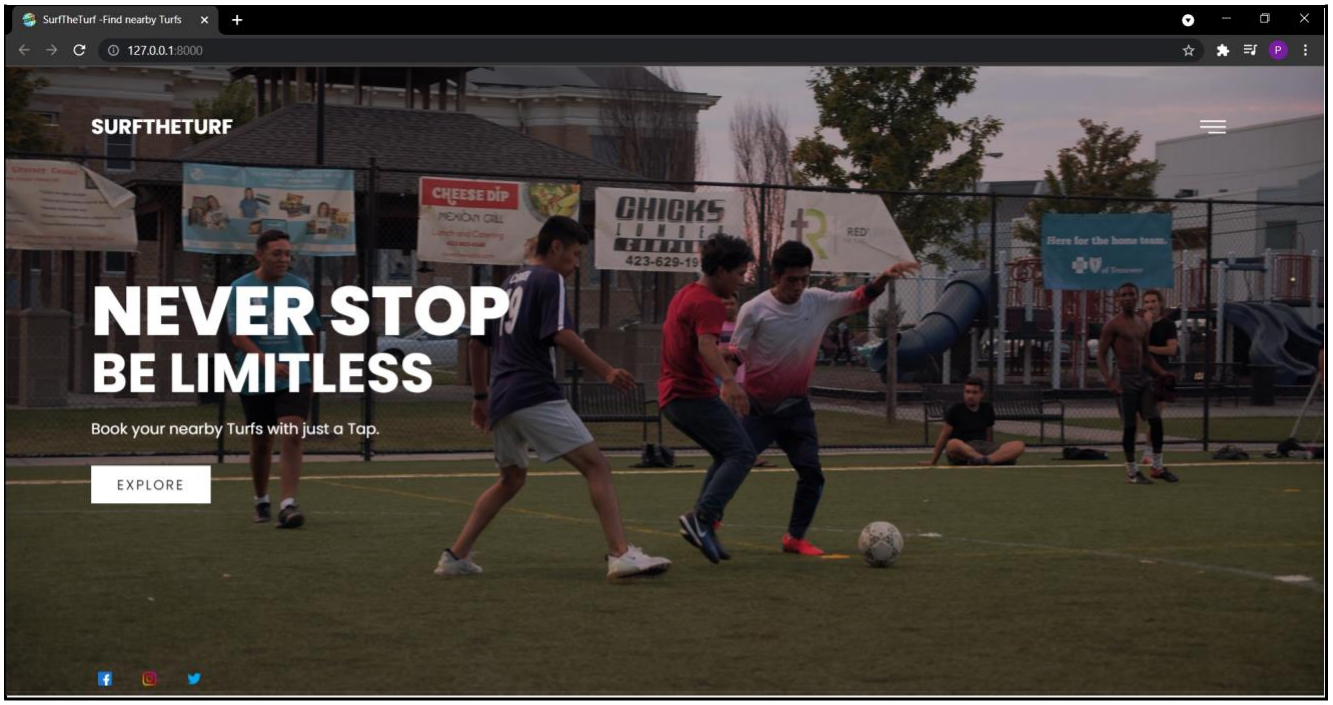
2. Backend :

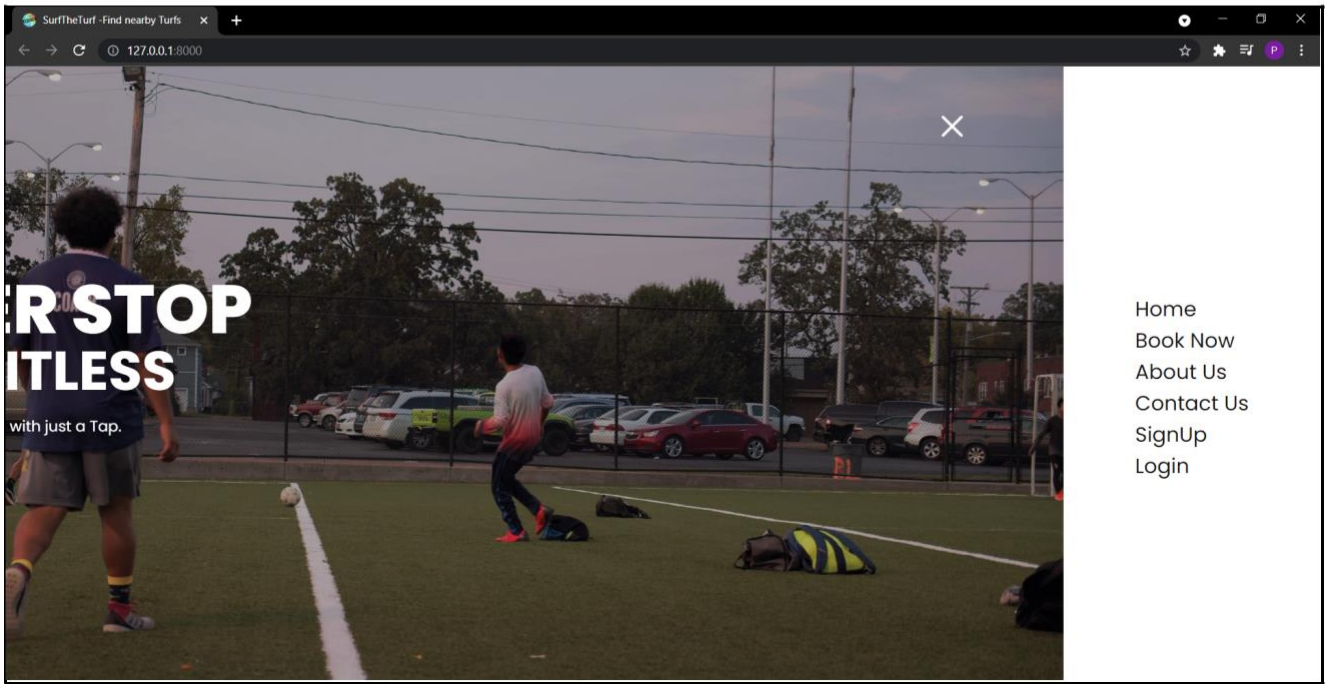
a. Django: We used it for slot booking, user signup, login and to connect different urls and views.

b. Postgre: We used Postgre to store the login credentials of our users and their booking details.

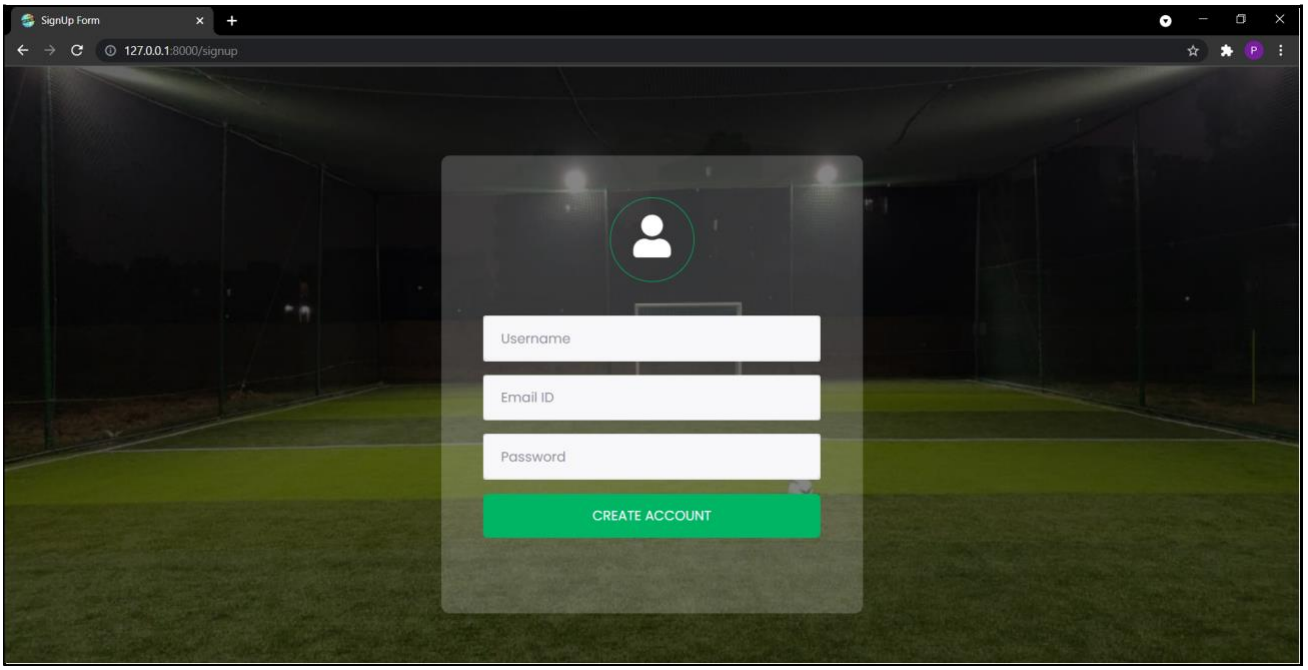
Results

Intro Page:

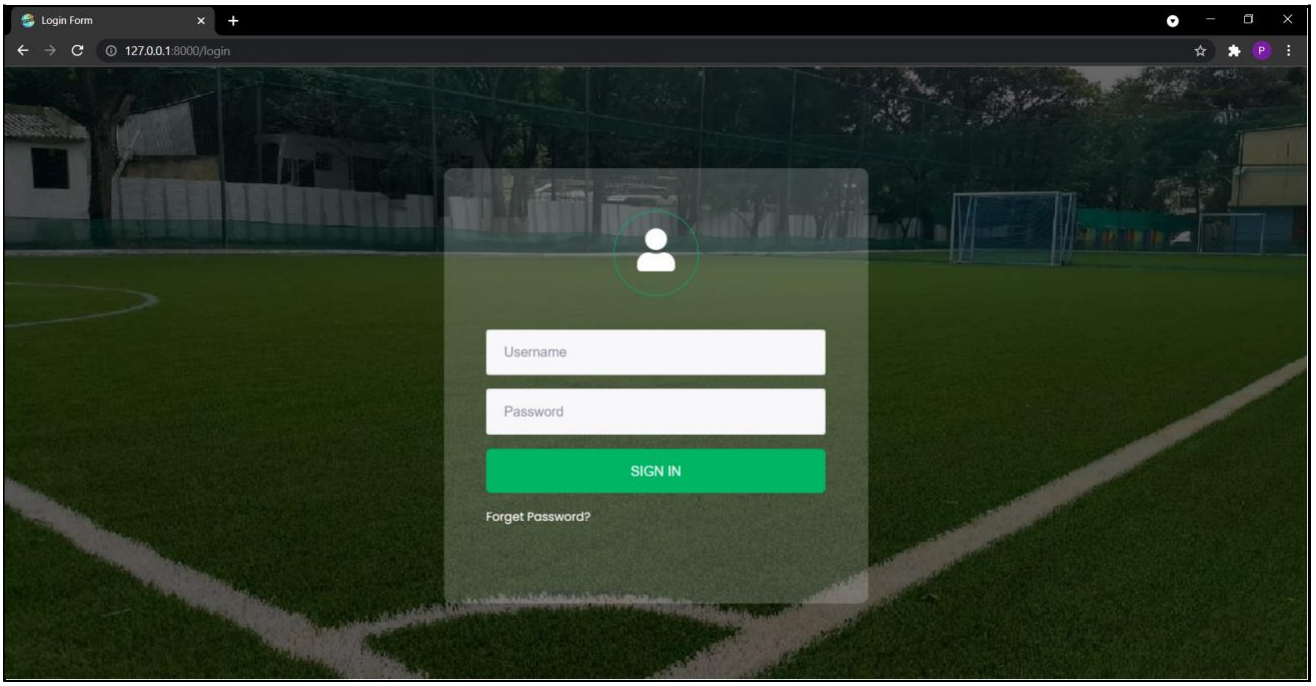




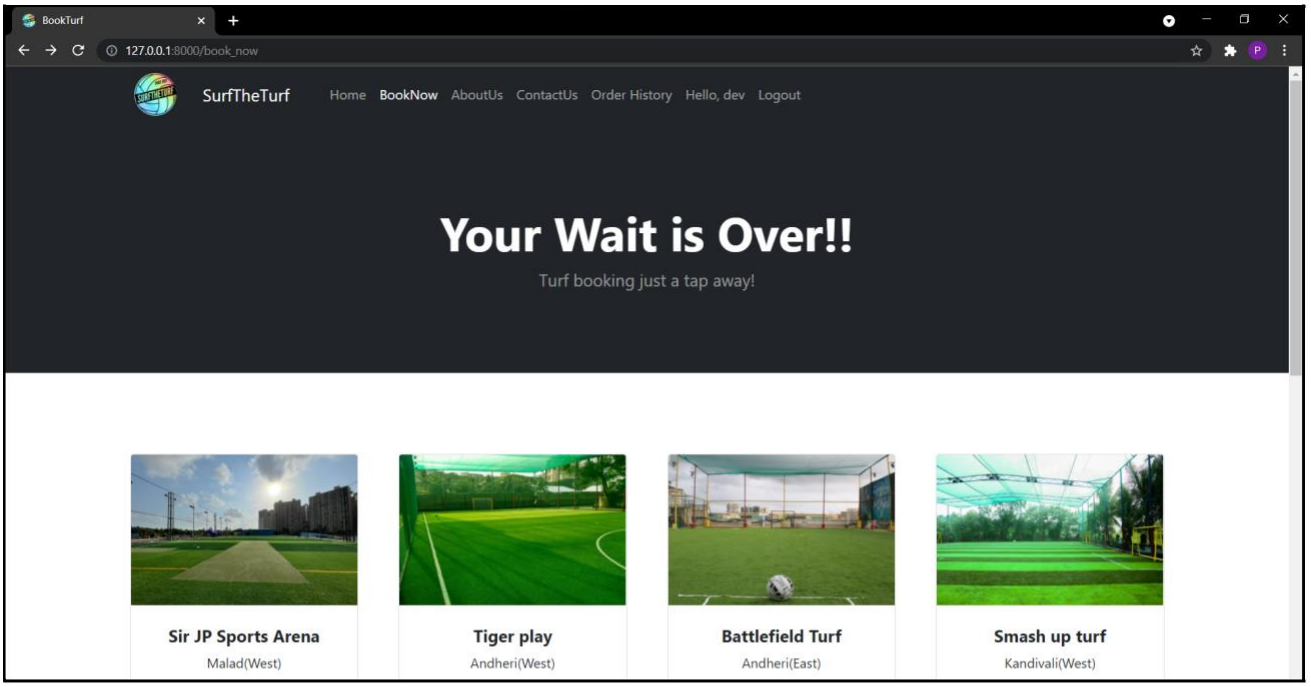
Sign Up Page:

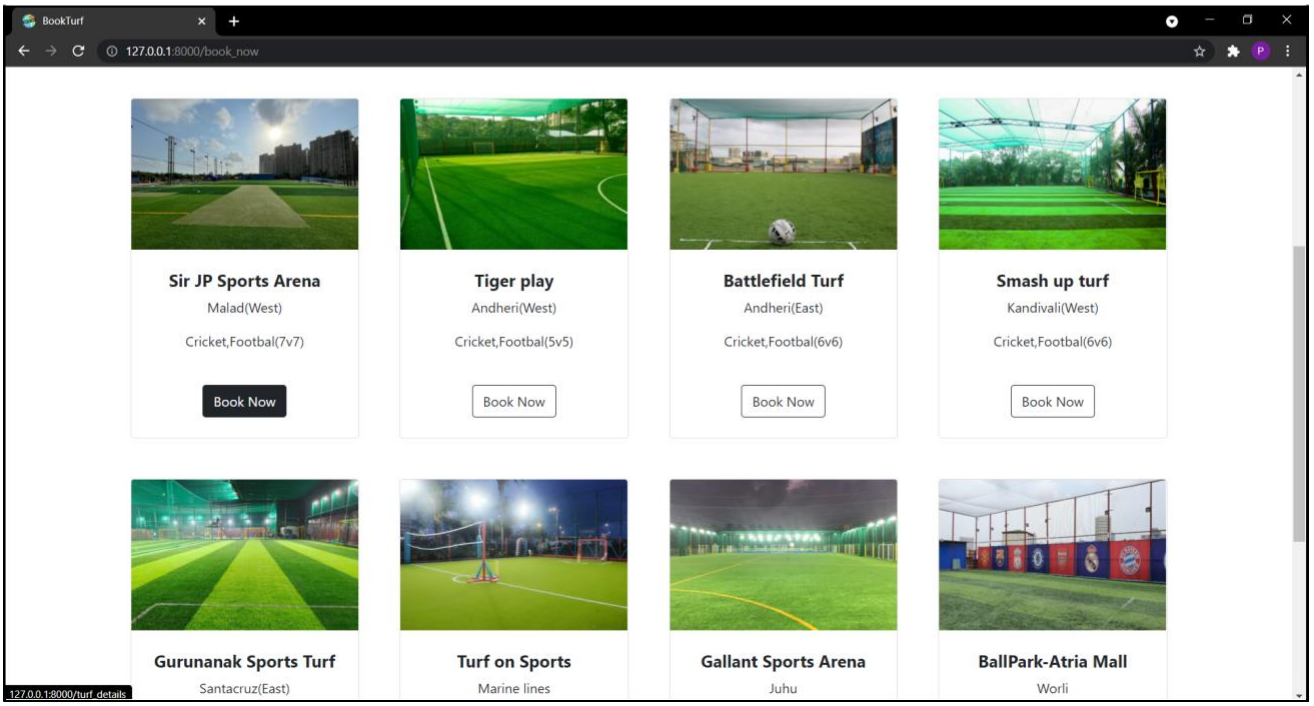


Login Page:

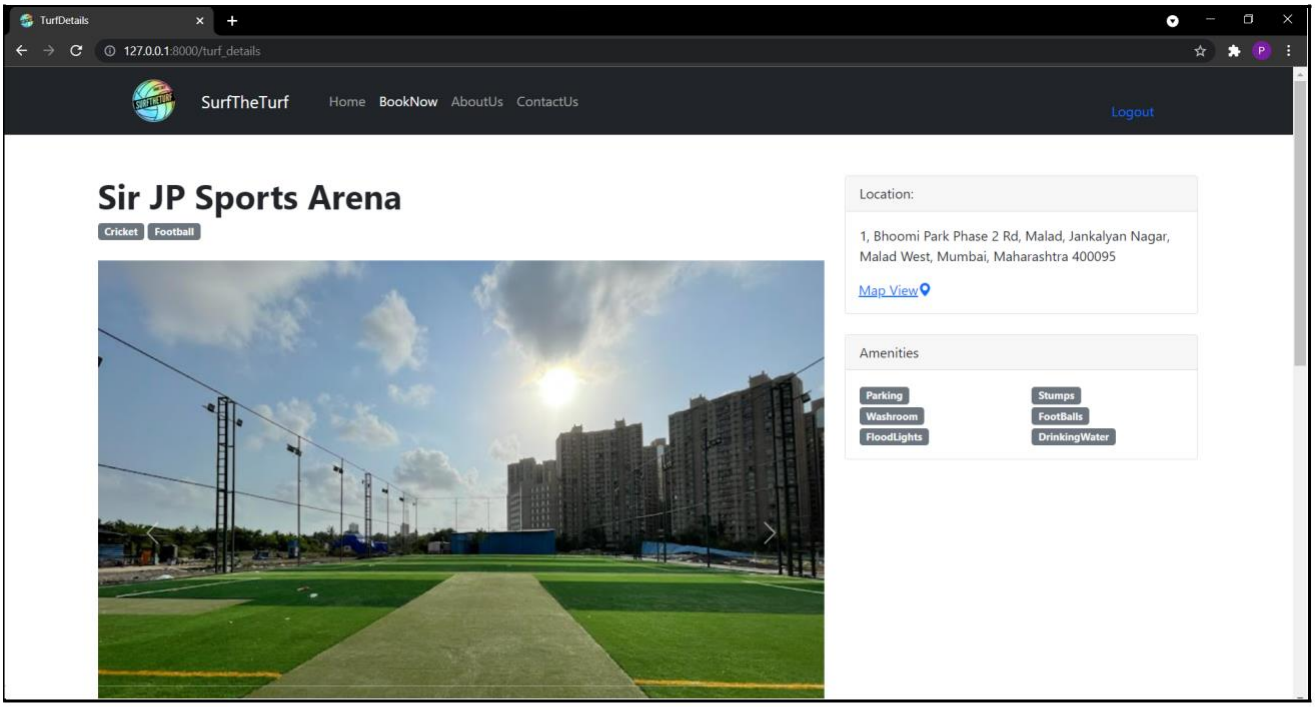


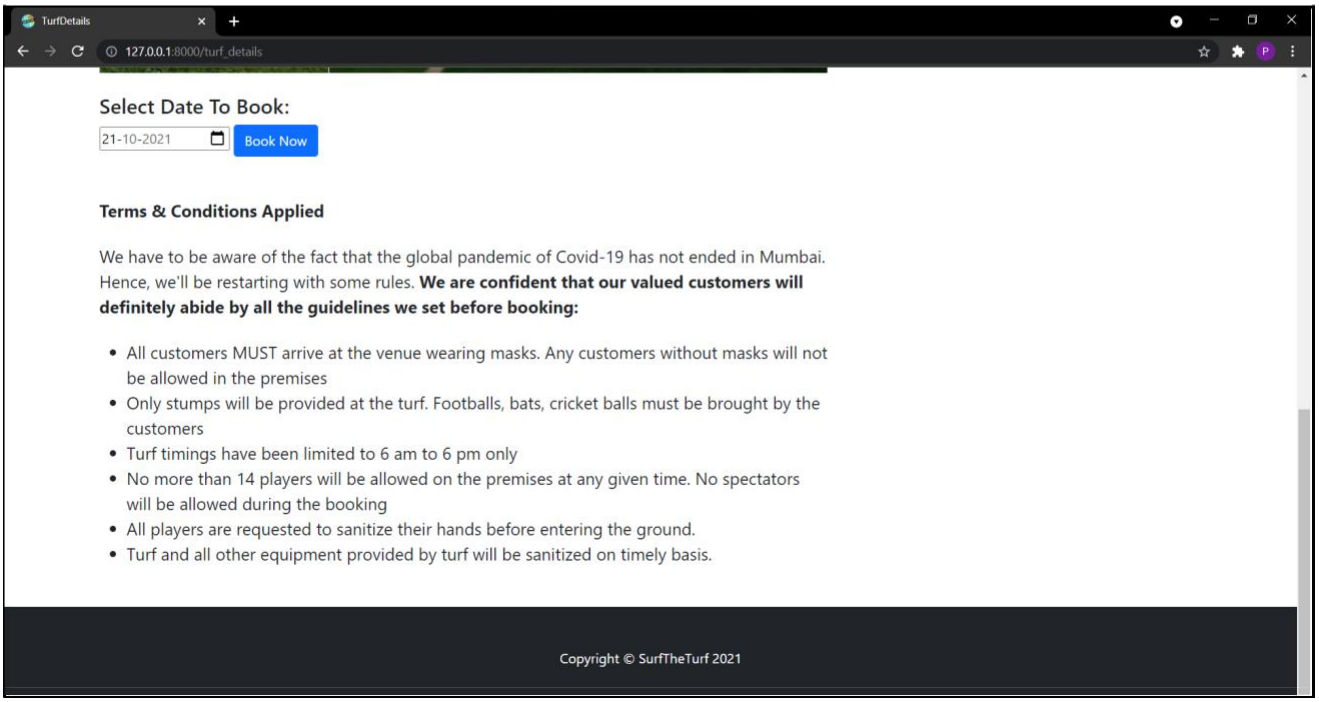
Booking Page:



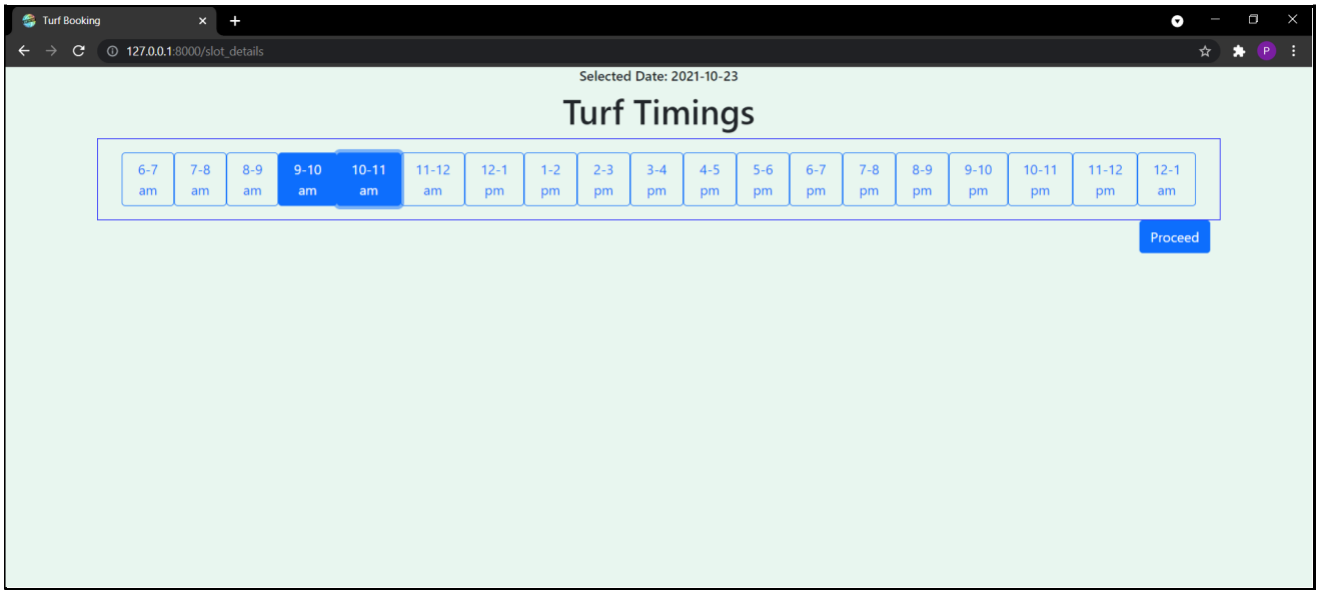


Turf Details Page:

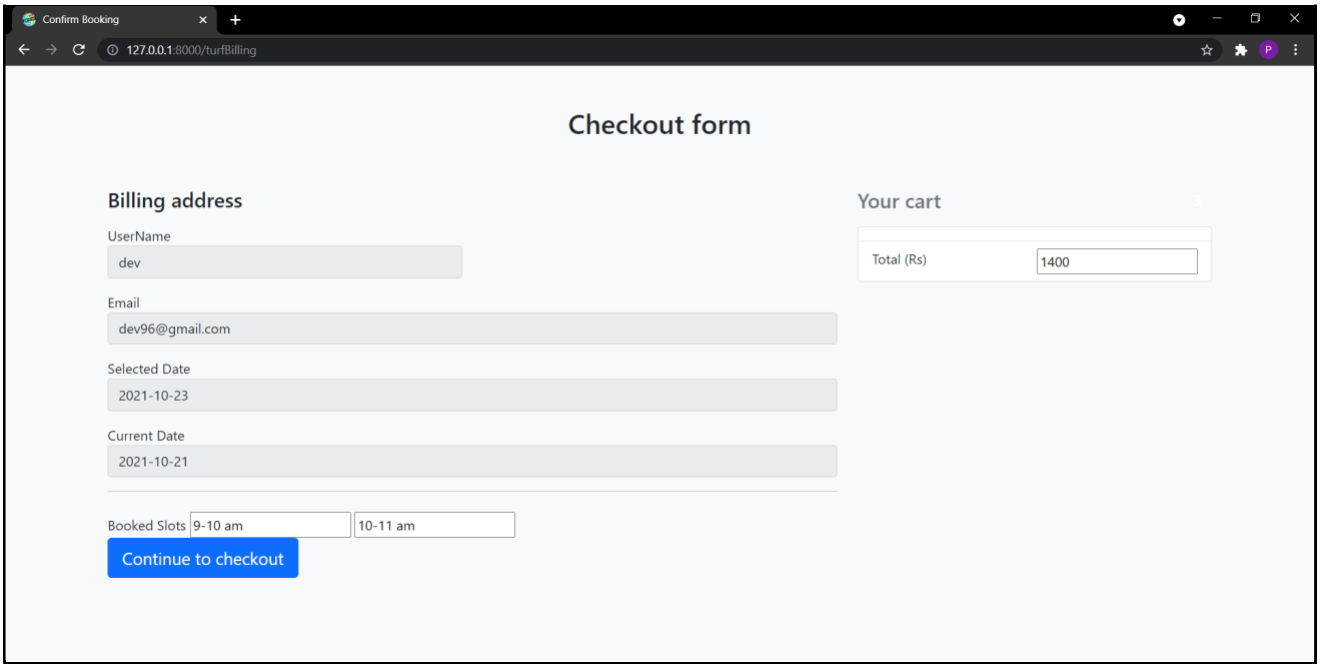




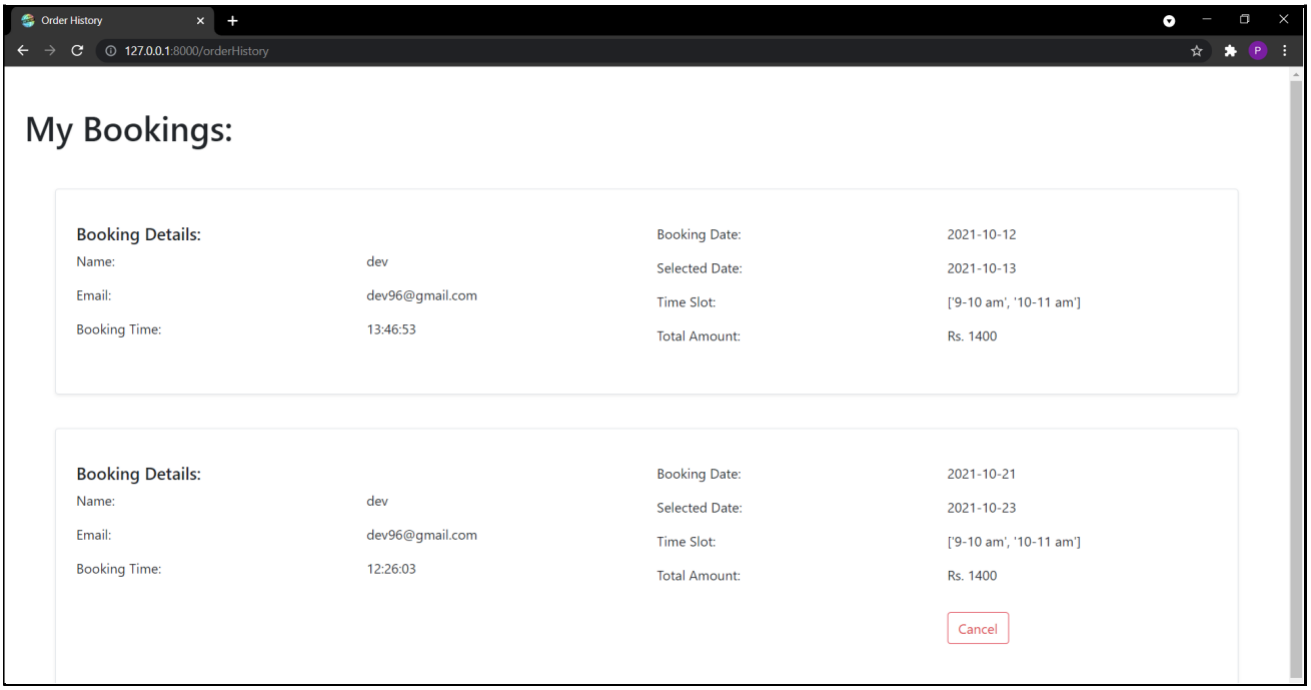
Slot Bookings Page:



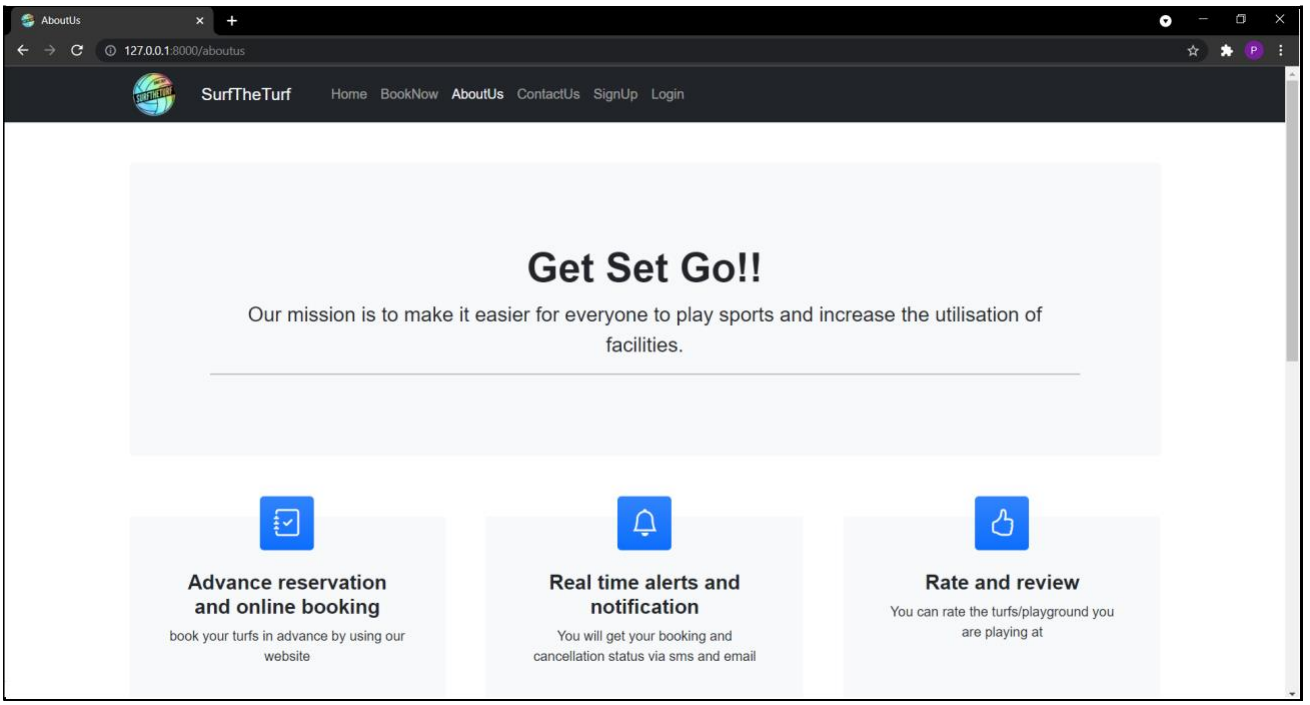
Checkout Form Page:



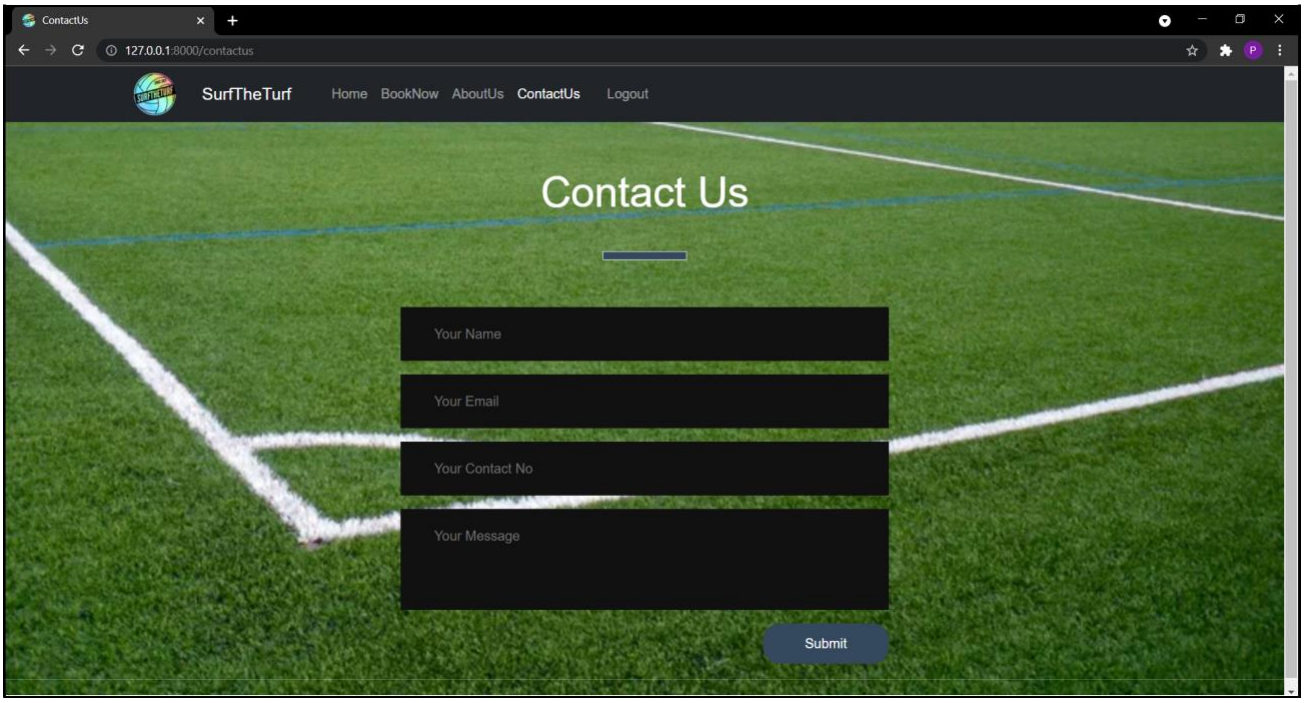
Order History:



About Us page:



Contact Us Page:



Challenges & Solutions

 Time Constraints:
Due to limited time, I first focused on implementing core functionalities like venue management, slot booking, and authentication. Features like payments and feedback were deferred for future development.

 Frontend-Backend Integration Issues:
During development, syncing the frontend forms with backend views proved challenging. Postman was used to test API endpoints, and error handling was added to ensure smooth communication.

 Database Schema Flexibility:
PostgreSQL was chosen for its robustness, but the schema required iterative changes to accommodate evolving requirements during development.

 Deployment Challenge – Rendering Images:
After deployment, an issue arose where venue photos and other images were not rendering properly. This was traced to incorrect static file configurations in the Django settings. The solution involved properly configuring the STATIC and MEDIA settings and ensuring that the web server (e.g., Nginx or Apache) served static and media files correctly.

Future Improvements

- **Admin Dashboard:**
  Build a dedicated dashboard for venue managers to monitor bookings and manage resources efficiently.
- **Payment Integration:**
  Add a secure payment gateway to allow users to complete their bookings with online payments.
- **User Feedback System:**
  Implement a rating and feedback system to gather user insights and improve the service quality.
- **Improving Scalability:**
  Introduce microservices architecture to separate key functionalities like notifications and user management, allowing the system to scale as the user base grows.

References

1.Dinesh A. Pol, Piyush V. Patil, Rahul Shinde, Rohan Dange " Online Ground Booking System using Android Mobile Application" International Journal for Scientific Research & Development Vol. 7, Issue 10, 2019.(Accessed: July 23, 2021)

2.AvailableOnline:https://www.fatbit.com/fab/category/web-based-business-ide as/ (Accessed: July 24, 2021)

3.AvailableOnline:https://softensy.com/how-to-create-payment-gateway-and-be come-a-payment-service-provider/(Accessed: July 25, 2021)