# Tennis Court Booking Web Application Development

by

Suleyman Eminov

Submitted to the Department of Mathematics and Computer Science

School of Natural and Social Sciences

in partial fulfillment of the requirements

for the degree of Bachelor of Arts

Purchase College

State University of New York

September 2024

Sponsor: Knarik Tunyan

Second Reader: [[Enter Second Reader’s name here]]

Table of Contents

[Tennis Court Booking Web Application Development 1](#_Toc179323257)

[Abstract 3](#_Toc179323258)

[1. Introduction 4](#_Toc179323259)

[2. Literature Review 4](#_Toc179323260)

[2.1. History of Agile Methodology 4](#_Toc179323261)

[2.2. Continuous Integration and Continuous Delivery (CI/CD) 4](#_Toc179323262)

[2.3. Docker 4](#_Toc179323263)

[3. Design and Architecture 6](#_Toc179323264)

[3.1. System Architecture 6](#_Toc179323265)

[4. Development Process 6](#_Toc179323266)

[4.1. Setup 6](#_Toc179323267)

[4.2. Docker 6](#_Toc179323268)

[4.3. Django 6](#_Toc179323269)

[4.4. Database 6](#_Toc179323270)

[4.5. React 6](#_Toc179323271)

[4.6. Integration 6](#_Toc179323272)

[5. Testing 6](#_Toc179323273)

[6. Conclusion 6](#_Toc179323274)

[Bibliography 7](#_Toc179323275)

[Appendix 8](#_Toc179323276)

# Abstract

This is a full-stack web development project using Django and React.

**Keywords**: Django, Python, JavaScript, React, Docker, PostgreSQL

# Introduction

To reserve a tennis court at a nearby tennis club, clients are currently required to contact the front desk and make a booking request. Unfortunately, there may be instances where the front desk is occupied, and the client's call goes unanswered. This presents an opportunity for the business to implement an online booking system, allowing clients to make reservations without relying on the front desk. This research aims to make this a reality. This research project aims to develop a full-stack web application that enables tennis clubs to manage their court time schedules and lets clients reserve a court remotely.

# Literature Review

The main question for this senior project is “What is the best way to develop a full-stack web application to enable tennis clubs to efficiently manage their court schedules and provide clients with a seamless remote reservation experience?”

## History of Agile Methodology

In 2001, a group of remarkable practitioners promoted an “Agile Manifesto.” The manifesto outlines the principles and values for developing software more flexibly and collaboratively, emphasizing customer satisfaction and continuous delivery. Since then, Agile methodologies have become increasingly popular in software development, with many teams adopting them as the standard approach. The Agile approach has been proven to increase productivity, improve quality, and reduce time-to-market for software products.

One of the popular frameworks of agile methodology is the Scrum methodology. It is an iterative and incremental approach to project management. Scrum breaks projects into manageable chunks called sprints which are time-boxed periods during which a team works on a set of specific tasks. This project will be using Scrum methodology to deliver functionality to the web app continuously.

## Continuous Integration and Continuous Delivery (CI/CD)

CI/CD is a set of best software development practices in which code changes are frequently and automatically built, tested, and deployed to production environments. This approach enables teams to deliver software quickly, reliably, and with minimal risk. By automating the build, test, and deployment processes, teams can reduce the time and effort required to release new features and fix bugs, while also improving the quality and stability of their software. One of the popular tools for CI/CD is known as GitHub Actions. Using GitHub Actions, we can define workflows that execute specific actions automatically whenever a push is made to a Git repository.

## Docker

Another important tool that will be used in this project is Docker. In software development, configuration discrepancies between your computer and the server on which your application is being run can lead to problems. For instance, you may have a different Python version, or additional packages installed on your computer that enable the application to run smoothly, while it would crash on the server. To circumvent these issues, it is necessary to ensure that everyone working on the project is using the same environment. Docker, a containerization software, can accomplish this by creating an isolated environment within your computer that can be standardized and shared among multiple collaborators and the server on which the site is hosted. It is worth noting that while Docker is similar to a virtual machine, they are distinct technologies. A virtual machine, such as the one used on GitHub Actions or AWS servers, is essentially an entire virtual computer with its own operating system, which can take up considerable space wherever it is running. Conversely, Docker operates by setting up a container within an existing computer, thereby occupying less space. (Adomnitei et al. *Web Portal Development with Different Cloud Containers: Docker vs. Kubernetes*)

The choice of using Django as the backend framework for this project is a wise one, as it is a powerful and flexible web framework that allows for quick and efficient development. Additionally, the use of React and JavaScript for the frontend will enable a seamless and intuitive user experience, as these technologies are known for their speed, interactivity, and scalability. Leveraging PostgreSQL as the database management system further enhances the capabilities of this tech stack, ensuring reliable and concurrent data management. Overall, this tech stack comprising Django, React, JavaScript, and PostgreSQL is a great choice for building a modern web application that is both robust and user-friendly.

Now, that we established the underlying tools that will be utilized in this project, let’s delve into the specifics of the web application. By the end of this project, we will have two parallel applications: one for businesses/admin and one for clients. The necessary features for the admin application are as follows:

* **Dashboard**: A centralized dashboard that displays real-time overviews of all the courts, reservations, and other pertinent information.
* **Court booking management**: The ability to add, edit, and delete courts, as well as set availability and schedule maintenance or other events.
* **Reservation management**: The ability to view, modify, and cancel reservations.
* **User management:** The ability to manage user accounts, including adding or removing users, setting permissions, and access levels.
* **Reporting and analytics**: The ability to generate reports and gather analytics on court usage, revenue, and other key metrics.

For the client application, the following features should be included:

* **Court search**: The ability to search for available courts based on location, date, time, and other criteria.
* **Court booking:** The ability to reserve a court for a specific date and time, and to view or modify existing reservations.
* **User account management**: The ability to create and manage user accounts, view past bookings, and update personal information.
* **Notification and reminders**: The ability to receive notifications and reminders about upcoming bookings, cancellations, and other important updates.

# Design and Architecture

Wireframes

## System Architecture

# Development Process

## Setup

How to setup this project

## Docker

What is docker. How it works? How docker is used in this projet?

## Django

Virtualenv

Django, djangorestframework

## Database

Postgres. Uml diagram

## React

## Integration



# Testing

Run Django tests.

# Conclusion

The main conclusions of the research may be presented in a short Conclusions section, which may stand alone or form a subsection of a Discussion or Results and Discussion section.

List funding sources if any:

This work was supported by the NSS Undergraduate Research [grant number XX, YEAR].

You may also include acknowledgements

Acknowledgements: …

**References/Bibliography**

Please ensure that every reference cited in the text is also present in the reference list (and vice versa). Reference list should follow the standard reference style, such as APA, MLA, etc. (Strang, 2016)

*To create References follow steps*

Document Elements -> Bibliography -> Manage

OR

References -> Citations

# Bibliography

Elsevier. (2021, September 15). *Guide for Authors, Linear Algebra and its Applications* . Retrieved from https://www.elsevier.com/journals/linear-algebra-and-its-applications/0024-3795/guide-for-authors

Strang, G. (2016). *Introduction to Linear Algebra.* Wellesley, MA: Wellesley-Cambridge Press.

1. Nampala, Henry, and Mahad Kigenyi. “A Response Concurrent Web-Based Hotel Reservation System: A Case Study of Sports View Hotel.” *Kampala International University*, July 2011, [ir.kiu.ac.ug/items/1217c92f-4f2b-4106-9e5f-3f37db73cdb8](http://ir.kiu.ac.ug/items/1217c92f-4f2b-4106-9e5f-3f37db73cdb8).
2. Adomnitei, Cezar, et al. “Web Portal Development with Different Cloud Containers: Docker vs. Kubernetes.” *Purchase College Library*, 2021, [suny-pur.primo.exlibrisgroup.com/discovery/fulldisplay?docid=cdi\_doaj\_primary\_oai\_doaj\_org\_article\_29ce7c61706447d6a965bbd244e84e13&context=PC&vid=01SUNY\_PUR%3A01SUNY\_PUR&lang=en&search\_scope=MyInst\_and\_CI&adaptor=Primo%2BCentral&tab=Everything&query=any%2Ccontains%2Cdocker%2BAND%2Bweb%2Bdevelopment&offset=0](http://suny-pur.primo.exlibrisgroup.com/discovery/fulldisplay?docid=cdi_doaj_primary_oai_doaj_org_article_29ce7c61706447d6a965bbd244e84e13&context=PC&vid=01SUNY_PUR%3A01SUNY_PUR&lang=en&search_scope=MyInst_and_CI&adaptor=Primo%2BCentral&tab=Everything&query=any%2Ccontains%2Cdocker%2BAND%2Bweb%2Bdevelopment&offset=0).
3. Fireship. “100+ Docker Concepts You Need to Know.” *YouTube*, YouTube, 12 Mar. 2024, [www.youtube.com/watch?v=rIrNIzy6U\_g](http://www.youtube.com/watch?v=rIrNIzy6U_g).
4. C.J., Torrecilla-Salinas, et al. “Estimating, Planning and Managing Agile Web Development Projects under a Value-Based Perspective.” *ScienceDirect*, 2015, [www-sciencedirect-com.ezproxy.purchase.edu/science/article/pii/S0950584915000142](http://www-sciencedirect-com.ezproxy.purchase.edu/science/article/pii/S0950584915000142).
5. <https://dev.to/anjalbam/dockerize-a-django-react-and-postgres-application-with-docker-and-docker-compose-by-anjal-bam-e0a>
6. <https://medium.com/@ronakchitlangya1997/jwt-authentication-with-react-js-and-django-c034aae1e60d>

# Appendix

Large tables and/or figures/many lines of code can be placed here.