

# Tennis Timer

Project Explaination

**Harikrishnan G**

Full Stack Developer || C# || React|| Typescript

harikrishnan14122000@gmail.com

9361513111

## Table of Content

S.NO	TOPIC	PAGE NO
1	Database(SQL)	3-4
2	Back-End	4-5
3	Front-End	5-11

## Overview

TennisTimer is a Webapp project created using React(Typescript Template) and ASP.NET Core 6.0 API. TennisTimer project showcases my knowledge in concepts like Entity framework, CORS, C#, SQL, Dependency Injection, Role Based authentication, API, handling images in database, React, React, HTML, CSS, Typescript, bootstrap. This project help the user to create a account to search the courts based on location and prebook with dates and slots. User information is protected using role based authentication system.

## Technology used in :-

1. ASP.NET CORE 6.0 (API)
2. HTML 5
3. Bootstrap 5.3.2
4. React 18.2.0

## Concepts Explanation:-

### Entity Framework:-

Entity Framework is a modern object-relation mapper that lets you build a clean, portable, and high-level data access layer with .NET (C#) across a variety of databases, including SQL Database (on-premises and Azure), SQLite, MySQL, PostgreSQL, and Azure Cosmos DB. It supports LINQ queries, change tracking, updates, and schema migrations.

### Dependency Injection:-

Dependency injection is used to make a class independent of its dependencies or to create a loosely coupled program. Dependency injection is useful for improving the reusability of code. Likewise, by decoupling the usage of an object, more dependencies can be replaced without needing to change class.

## Database(SQL):-

Database for this project is created using SQL language which include seven tables. Three tables stores the data of users like User details, Roles, UserRoles. One table stores the Refresh Token. Another three table stores the Court Information, Court Booking Information, Court User Review Information.

User table is connected with UserRole, Refresh Token, Court Booking and Court Review with ON DELETE CASCADE relationship. Court Table is connected using Court Booking, Court Review. Below is the image of SQL Code

```

Create DATABASE TennisTimer;

Create TABLE Users(
    UserName VARCHAR(20) PRIMARY KEY ,
    PasswordHash VARBINARY(MAX) NOT NULL,
    PasswordSalt VARBINARY(MAX) NOT NULL
);

Create Table Roles(
    RoleId int PRIMARY KEY,
    RoleName Varchar(20) NOT NULL
);

Create Table UserRole(
    UserName VARCHAR(20) PRIMARY KEY,
    RoleId int NOT NULL,
    FOREIGN KEY (UserName) REFERENCES Users(UserName) ON DELETE CASCADE,
    FOREIGN KEY (RoleId) REFERENCES Roles(RoleId) ON DELETE CASCADE
);

Create Table Court (
    CourtId INT IDENTITY(1,1) PRIMARY KEY,
    CourtName VARCHAR(20) NOT NULL,
    CourtLocation VARCHAR(50) NOT NULL,
    CourtImg1 VARBINARY(MAX),
    CourtImg2 VARBINARY(MAX),
    CourtImg3 VARBINARY(MAX)
);

Create Table CourtBooking(
    BookingID INT IDENTITY(1,1) PRIMARY KEY,
    CourtId INT NOT NULL,
    BookingDate DateTime NOT NULL,
    Slot1 INT,
    Slot2 INT,
    Slot3 INT,
    UserName VARCHAR(20) NOT NULL
    FOREIGN KEY (CourtId) REFERENCES Court(CourtId) ON DELETE CASCADE,
    FOREIGN KEY (UserName) REFERENCES Users(UserName) ON DELETE CASCADE
);

```

```
Create Table CourtReview(
    ReviewId INT IDENTITY(1,1) PRIMARY KEY,
    CourtId INT NOT NULL,
    UserName VARCHAR(20) NOT NULL,
    BookingID INT NOT NULL,
    Rating INT
    FOREIGN KEY (CourtId) REFERENCES Court(CourtId) ON DELETE CASCADE,
    FOREIGN KEY (UserName) REFERENCES Users(UserName) ON DELETE CASCADE,
    FOREIGN KEY (BookingID) REFERENCES CourtBooking(BookingID)
);
```

## Back-End (ASP.NET.CORE API):-

Backend is created using ASP.NET Core 6.0 API where I implemented Dependency Injection, Entity Framework, Swagger, Role based authentication, Image to byte array function to store images in SQL server. Below is the image of API swagger.

The screenshot shows the Swagger UI interface for the TennisTimer API. At the top, there's a navigation bar with the Swagger logo, a dropdown for 'Select a definition' set to 'TennisTimer v1', and a 'Customize and control Google Chrome' button. Below the header, the title 'TennisTimer 1.0 (OAS3)' is displayed, along with the URL 'http://localhost:5194/swagger/v1/swagger.json'. A green 'Authorize' button is located on the right side of the header. The main content area is organized into sections: 'Auth', 'CourtBookings', and 'CourtReviews'. Each section contains a list of API endpoints with their methods and URLs. The 'Auth' section has three POST methods: '/api/Auth/register', '/api/Auth/login', and '/api/Auth/Refresh'. The 'CourtBookings' section has five methods: GET '/api/CourtBookings', POST '/api/CourtBookings', GET '/api/CourtBookings/{id}', PUT '/api/CourtBookings/{id}', and DELETE '/api/CourtBookings/{id}'. The 'CourtReviews' section also has five methods: GET '/api/CourtReviews', POST '/api/CourtReviews', GET '/api/CourtReviews/{id}', PUT '/api/CourtReviews/{id}', and DELETE '/api/CourtReviews/{id}'.

Auth	
POST	/api/Auth/register
POST	/api/Auth/login
POST	/api/Auth/Refresh

CourtBookings	
GET	/api/CourtBookings
POST	/api/CourtBookings
GET	/api/CourtBookings/{id}
PUT	/api/CourtBookings/{id}
DELETE	/api/CourtBookings/{id}

CourtReviews	
GET	/api/CourtReviews
POST	/api/CourtReviews
GET	/api/CourtReviews/{id}
PUT	/api/CourtReviews/{id}
DELETE	/api/CourtReviews/{id}

The screenshot shows the Swagger UI interface for a RESTful API. It is organized into two main sections: 'Courts' and 'Users'. The 'Courts' section contains the following endpoints:

- Courts** (Group Title):
  - GET /api/Courts**
  - POST /api/Courts**
  - GET /api/Courts/{id}**
  - PUT /api/Courts/{id}**
  - DELETE /api/Courts/{id}**
  - POST /api/Courts/PostCourt**

The 'Users' section contains the following endpoints:

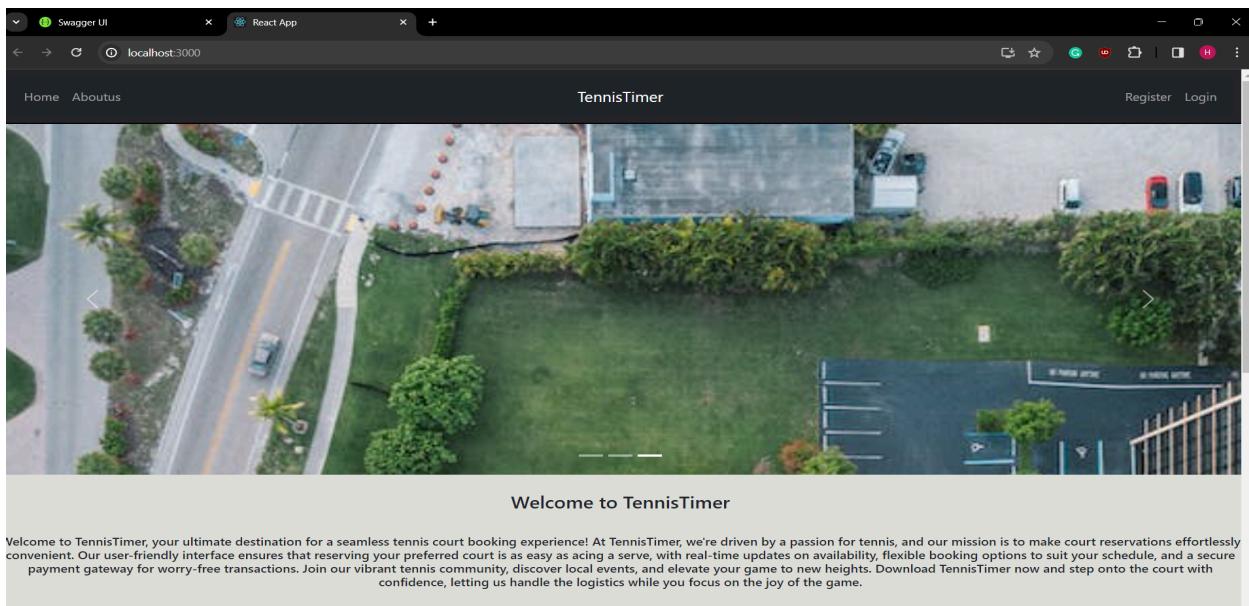
- Users** (Group Title):
  - GET /api/Users/GetUser**
  - GET /api/Users/ GetUserRole**
  - GET /api/Users/GetRoles**
  - POST /api/Users/MapRoles**
  - DELETE /api/Users/{id}**

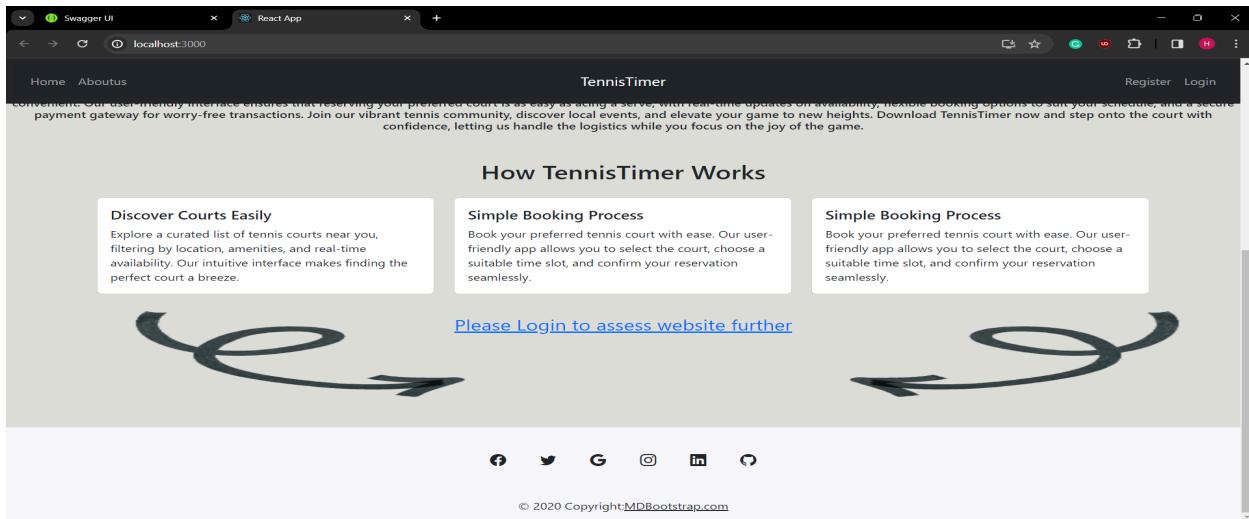
## Front-End(React Typescript):-

Front-end of TennisTimer is created using React 18.2.0 with typescript template, it was connected using CORS with ASP.NET CORE API. The flow of the website is given below.

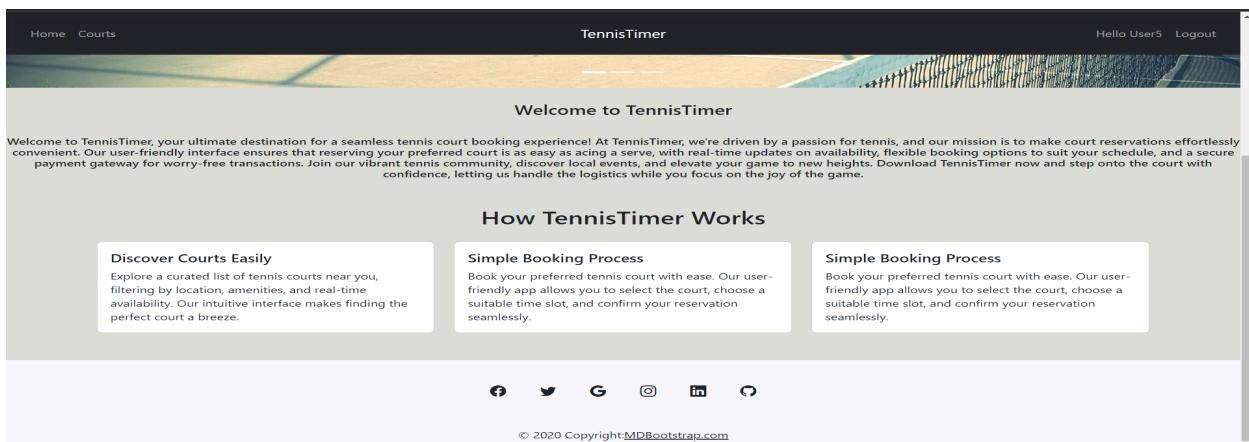
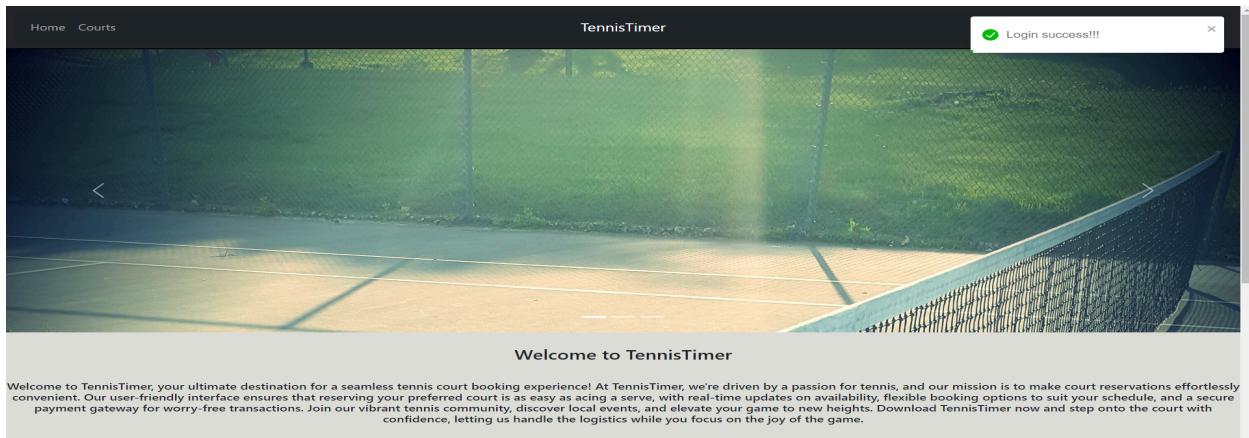
### Home Page:-

Home page of the website before login, it has navbar with links for Home page, About Page, Registration Page, Login Page. At bottom of the page If users didn't login it has a message to login to see further information





Once Userloggedin message disappear and new links appear in navbar for Courts SearchPage, User Profile page and Logout link



TennisTimer has a fixed position navbar at the top.

## Courts Search:-

Court Search page as a table with search window on the top to select the location, where they want to book the court. Court Location Data was fetched from the database to display in the dropdown

The screenshot shows a web browser window with the URL `localhost:3000/CourtSearch`. The page title is "TennisTimer". The main content area is titled "Court List". It features a table with columns "S.No", "Court Name", and "Court Location". Above the table is a search bar labeled "Search" and a dropdown menu labeled "Select Location" containing the following options: Kelavasal, Periyar, ByPass, Aarapalayam, periyar, and ByPass. The "Periyar" option is highlighted.

The screenshot shows the same web browser window after a location has been selected. The dropdown menu is no longer visible. The search bar now contains the text "Kelavasal". The table now displays a single row of data: S.No 1, Court Name Court 4, and Court Location Kelavasal. There is a "View" button next to the last column.

Once location is selected, the table displays the courts present in the location.

## Court View:-

Court View page has the information about the court like name, images of the courts given while register the court, overall rating of the court and Link to book the court.

↑ Preview of the court Above ↑

**Court Details**

Name of the court is : Court 4

Court Located in : Kelavasal

Overall Rating : 4.5

[Back](#) || [Book](#)

## Court Booking:-

If user want to book the court they have to provide the date and slot for the booking purpose.

↑ Preview of the court Above ↑

**Court Booking**

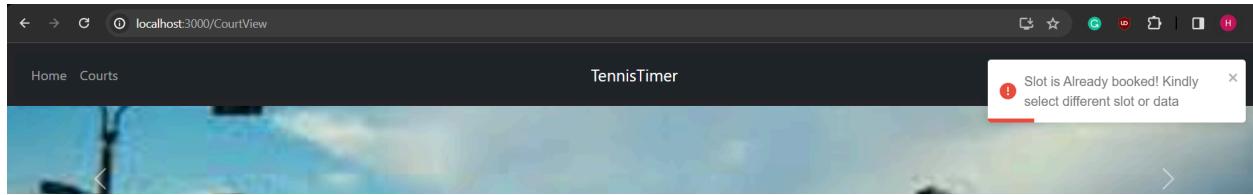
Enter date  
12-01-2024

Select Slot  
Slot 1 - 6:00 Am to 8:00AM

[Book](#) [Cancel](#)

Once user booked the court, they will be navigated to the user profile page which has there booking history

If User booked the slot which was already booked by other user, they will get the notification to book another slot.



## User Profile:-

User Profile page has the user booking history and have a link to place reviews on the court they already booked

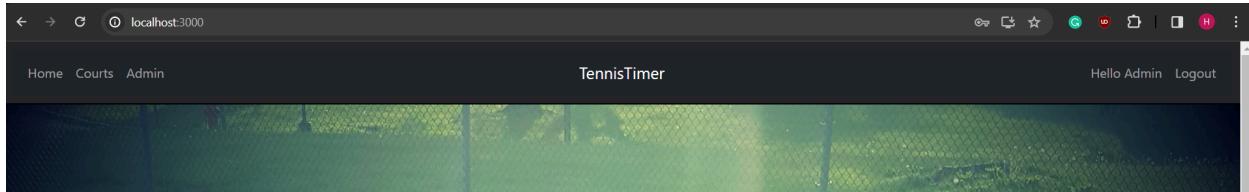
A screenshot of a web browser window titled 'localhost:3000/UserProfile'. The header shows 'Home', 'Courts', 'TennisTimer', 'Hello User5', and 'Logout'. The main content area features a large 'Hi User5 !!!' greeting. To its right is a box labeled 'Number of booking' containing the number '3'. Below this is a section titled 'User5 bookings below' containing a table of booking history:

S.No	Booking ID	Court ID	Booking Date	Slot No 1	Slot No 2	Slot No 3	Booked User	
9	27	4	2024-01-17T00:00:00	Free	Booked	Free	User5	<button>Add Rating</button>
10	28	4	2024-01-12T00:00:00	Booked	Free	Free	User5	<button>Add Rating</button>
11	29	4	2024-01-17T00:00:00	Booked	Free	Free	User5	<button>Add Rating</button>

A screenshot of a web browser window titled 'localhost:3000/UserProfile'. The header shows 'Home', 'Courts', 'TennisTimer', 'Hello User5', and 'Logout'. A modal window titled 'Booking Review' is open in the center. It contains a rating scale labeled 'Rate us: [ ]', a 'Submit Rating' button, and a 'Cancel' button. The background of the page is dark gray with the text 'Hi User5 !!!' visible.

## Admin Page:-

Admin page link was only available to the “Admin” role user account in the navbar.



Admin page has the all User Information, User Count Court Information, Court Count, Booking Information, Review Information. Admin can delete the users and Courts, Admin can give another user admin access using UI.

S.No	Name	
1	Admin	<button>Delete</button>    <button>Edit</button>
2	User	<button>Delete</button>    <button>Edit</button>
3	User1	<button>Delete</button>    <button>Edit</button>

S.No	Court Name	Court Location	+ Add Court
1	Court 4	Kelavasal	<button>Delete</button>    <button>View</button>
2	Court 5	Periyar	<button>Delete</button>    <button>View</button>
3	Court 6	ByPass	<button>Delete</button>    <button>View</button>
4	Court 8	Aarapalayam	<button>Delete</button>    <button>View</button>
5	Court 9	periyar	<button>Delete</button>    <button>View</button>
6	Court 10	ByPass	<button>Delete</button>    <button>View</button>

S.No	Booking ID	Court ID	Booking Date	Slot No 1	Slot No 2	Slot No 3	Booked User
1	3	11	2024-01-05T00:00:00	Free	Booked	Free	Admin

The screenshot shows a web application interface for 'TennisTimer'. At the top, there is a navigation bar with links for 'Home', 'Courts', 'Admin', 'Hello Admin', and 'Logout'. Below the navigation bar is a title 'TennisTimer'.

**Booking History Table:**

S.No	Court ID	User ID	Date	Status	Booking ID	Free/Court ID	Free/Court ID	Role
5	19	11	2024-01-14T00:00:00	Booked		Free	Free	Admin
6	20	11	2024-01-14T00:00:00	Free		Booked	Free	Admin
7	21	4	2024-01-14T00:00:00	Booked		Free	Free	User
8	26	4	2024-01-16T00:00:00	Booked		Free	Free	User4
9	27	4	2024-01-17T00:00:00	Free		Booked	Free	User5
10	28	4	2024-01-12T00:00:00	Booked		Free	Free	User5
11	29	4	2024-01-17T00:00:00	Booked		Free	Free	User5

**Review Data Table:**

S.No	Review ID	Court ID	Username	Booking ID	Rating
1	1	11	Admin	3	4
2	2	11	Admin	3	2
3	6	4	User4	26	4
4	7	4	User5	27	5

The screenshot shows a modal dialog box titled 'Edit User' over a background page displaying 'Number of Users' (5) and 'User Info' (11).

**Edit User Dialog:**

Assign the role to the : **User1**

**Submit** **Cancel**

## Conclusion:-

This project showcase my skills in above concepts and technology. I am eager to use my knowledge in different project and learn new concepts and technology