Shaakya Weerasundera (s307943) and Eric Heng (s322698)

  [Company address]

HIT339 – Distributed Development – Assignment 2 Documentation

Contents

[Web App Overview 2](#_Toc53483276)

[How to use the app 2](#_Toc53483277)

[How we made the app 2](#_Toc53483278)

[Making the app Mobile Ready: 3](#_Toc53483279)

[Problems encountered and how they were solved: 15](#_Toc53483280)

[References 16](#_Toc53483281)

# Web App Overview

It was requested to build an ASP .NET MVC web application for a tennis company. Using Visual Studio Community 2019, the team developed an ASP .NET web application for users with multiple privileges. The web app can be accessed by members wanting to join the tennis club and coaches of the tennis club. In this project, a user with admin privileges has the ability to use the application to a higher level. The functionality of the app is follows. A member wishing to join the tennis club has the ability to register as a member. Once registered, the member can log in to the account created and view the current coaches conducting events in the club. Members also have the ability to view upcoming events and schedules. When a coach logs in to the app, they will be able to view other coaches in the club, view all the events and schedules. In addition to this, the coach will also have the ability to edit their details in their profile. When a user with admin privileges logs in to the app, they will have access to a lot. They will be able to view all roles (such as Admin, Coach and Member). They will also be able to create new roles, view the roles, edit and delete. In addition to this, the admin also has the ability to view all coaches in a list, see coach details, edit coach details and delete existing records. In the members page, the admin will be able to see all member details, edit and delete records of members. In addition, they will also have the ability to go to the Events page and Create a new event. Admin will be able to view event details, edit and delete events. Finally, they will also have the ability to go to the All Schedules pages and view details of upcoming schedules.

The team used Github to commit changes to the web application.

The Github link for the app is here: <https://github.com/erichengyc/WebApplication1>.

# How to use the app

* Click on the WebApplication1.sln and open it in Visual Studio Community
* Run the Update-Database migration to make sure the database is updated
* Ctrl-f5 to run the application.
* You land in home page
* **Members:**
  + Go to the Register page
  + Enter your details
  + Click Register button
  + Go to Login page
  + Enter your credentials and click Login
  + You can now view coaches, events, schedules and your profile.
* **Coaches:**
  + Login with username [coach@coach.com](mailto:coach@coach.com) and password coach.
  + You can now view coaches, events, schedules and edit your profile
* **Admin:**
  + Login with username [admin@admin.com](mailto:admin@admin.com) and password admin.
  + Go to Roles, Coaches, Members, Events, Schedules pages.
  + Test the Create, Edit, Details and Delete pages of the Entities.
  + You can create new Roles and Events.
  + You can also delete existing roles members, coaches and events.

# How we made the app

* Create the models for the Event
* Add the EventsController
* Create the Member model and link to Event
* Add the corresponding controller MembersController
* Create the Schedule model
* Add the SchedulesController
* Create the Role model
* Add the RolesController
* Create the tennisContext and add a Connection String in the appsettings.json file to the database WebApplication3
* Seed the data in the SeedData.cs file.
* Use scaffolding to create the Create, Read, Update, Delete (CRUD) functionality.
* Add the necessary migrations
* The team used the Visual Studio Community and committed the code remotely via Github. Code was constantly committed, synced, and pushed to make the app up to date with the latest code.
* The team used manual authorization and authentication to create the coach, member and admin accounts. After they were created, the logins were used to test the functionality of the app. The credentials used were as follows:

1. Member: CDU username and password
2. Coach: username: [coach@coach.com](mailto:coach@coach.com). Password: coach
3. Admin: username: [admin@admin.com](mailto:admin@admin.com). Password: admin

* Use Fonts Awesome CSS to style the application and add cards to make the app look mobile friendly.

Note: To do the code, the team followed the tutorials located in the following link in the Microsoft ASP .NET documentation:

<https://docs.microsoft.com/en-us/aspnet/core/data/ef-mvc/intro?view=aspnetcore-3.1>

<https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/adding-model?view=aspnetcore-3.1&tabs=visual-studio>

We used CSS cards by following the W3Schools CSS cards tutorial:

<https://www.w3schools.com/howto/howto_css_cards.asp>

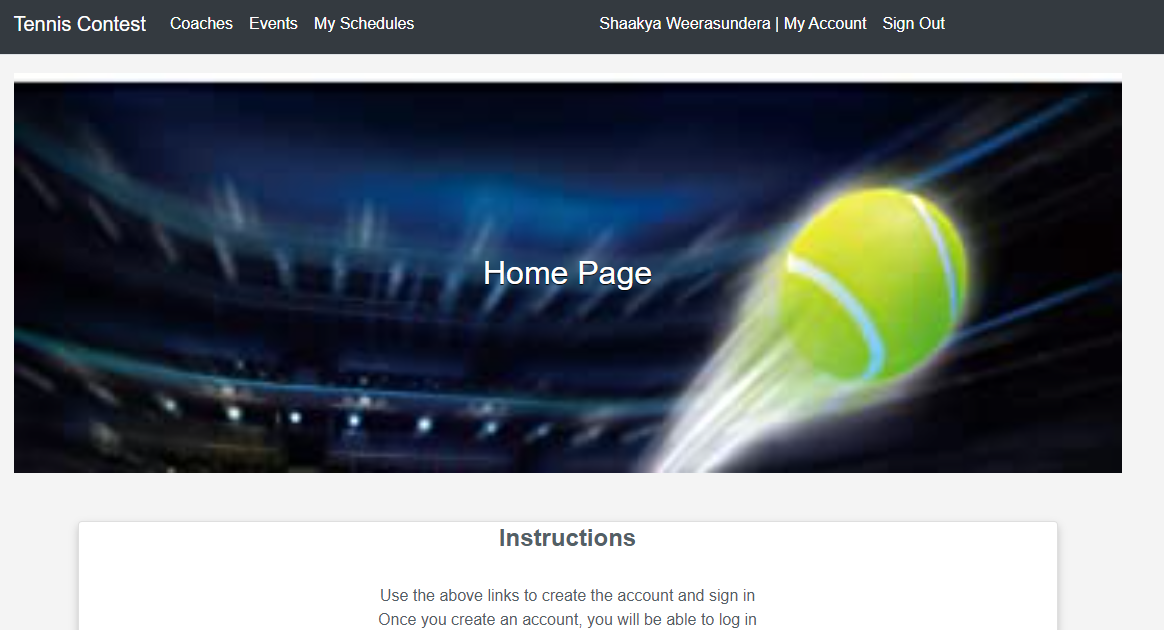
# Making the app Mobile Ready:

The team decided to make the app mobile ready. Using CSS tricks, we styled the application to make it look nicer. On a desktop view, the pages of the app are as follows:

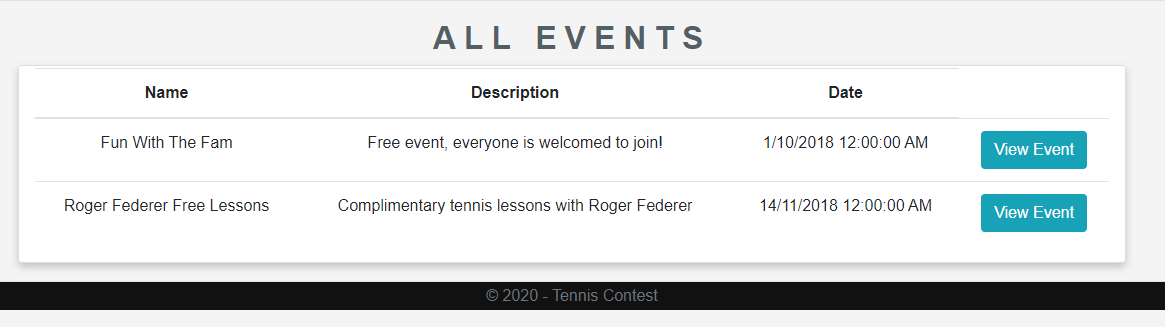
In Member account:

* Members can view schedules with the coaches coaching the event
* See coach profiles from a list
* Sign up for an account and log in so that an email can be sent to notify them of schedules.
* Members can also enrol in a schedule before the date scheduled.

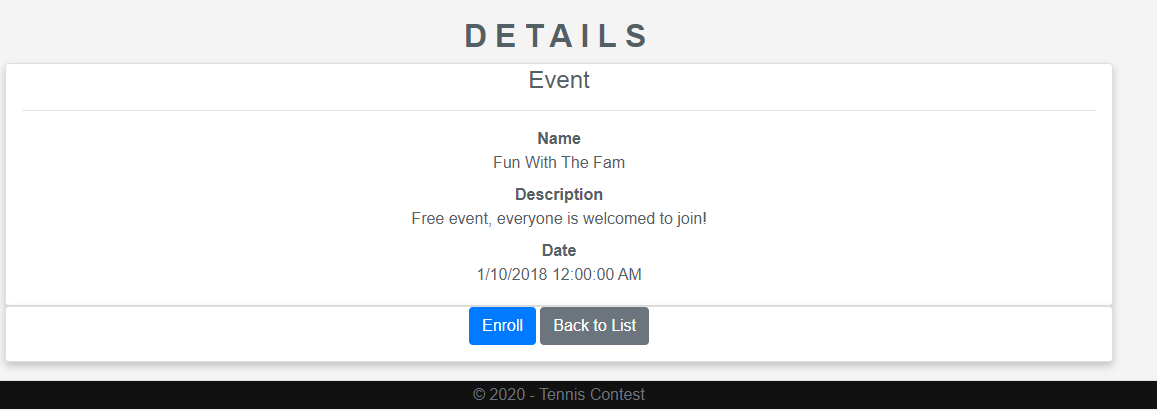
Home Page



Events Page



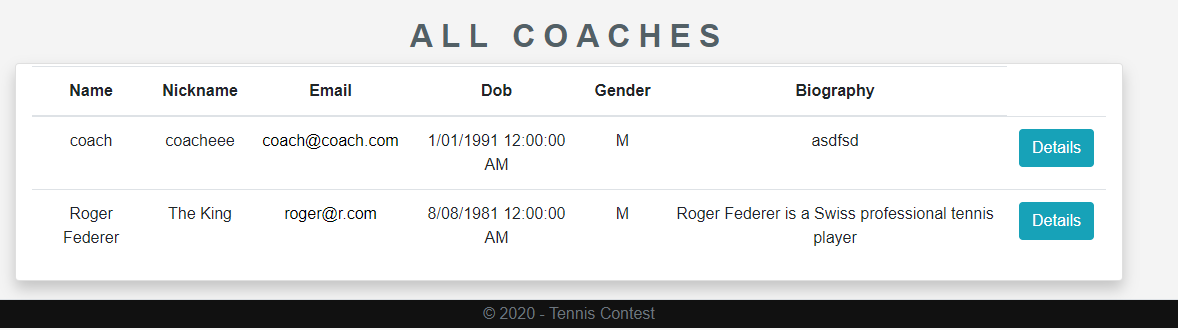
Events Enroll Page



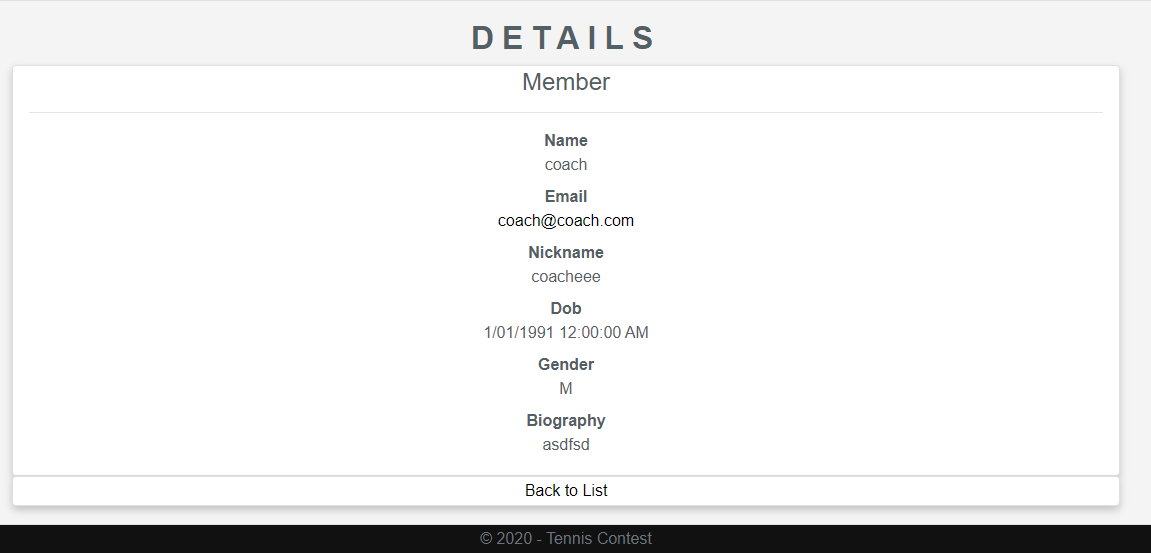
Coach Account:

* Coaches can Create and Edit the biography of a coach profile
* Can also see upcoming schedules
* View enrolled members for schedules.
* Coaches do not have any additional privileges.

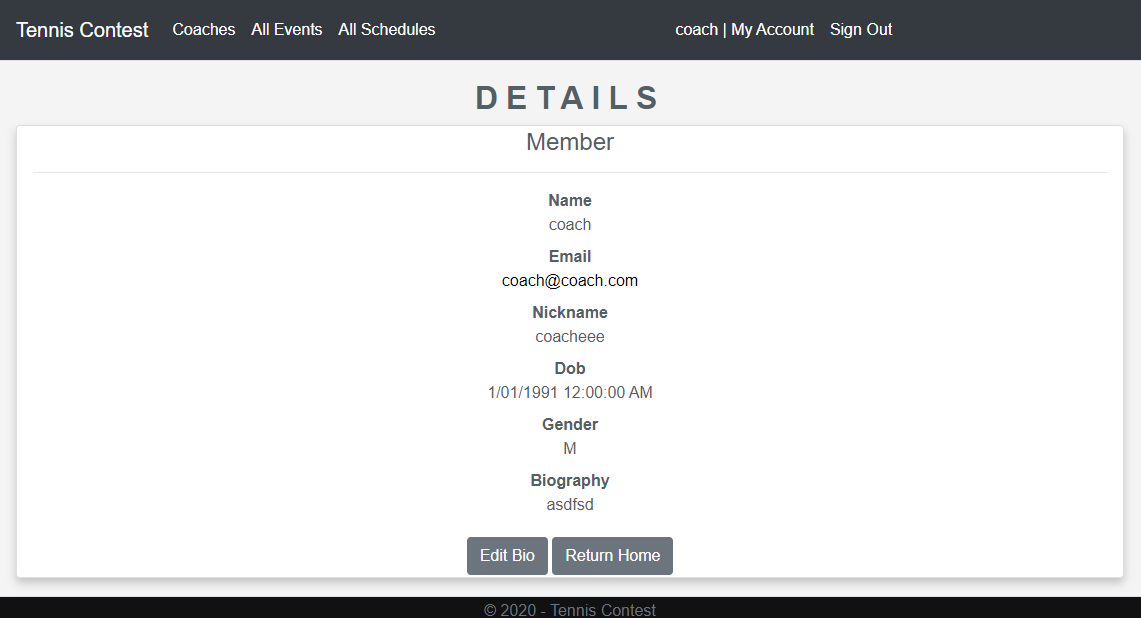
Coach Page



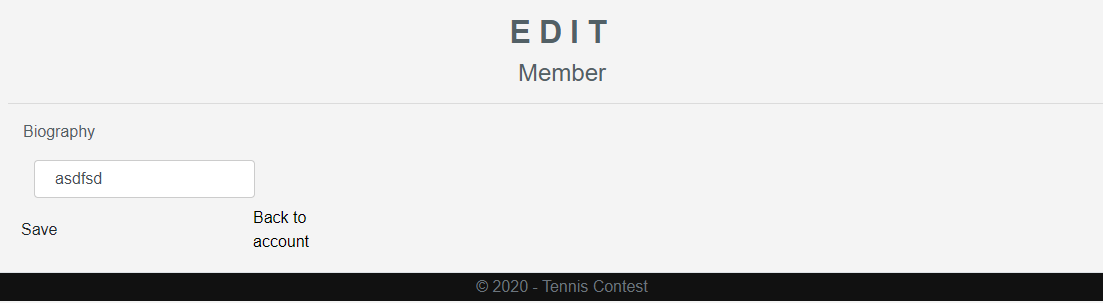
Coaches Details



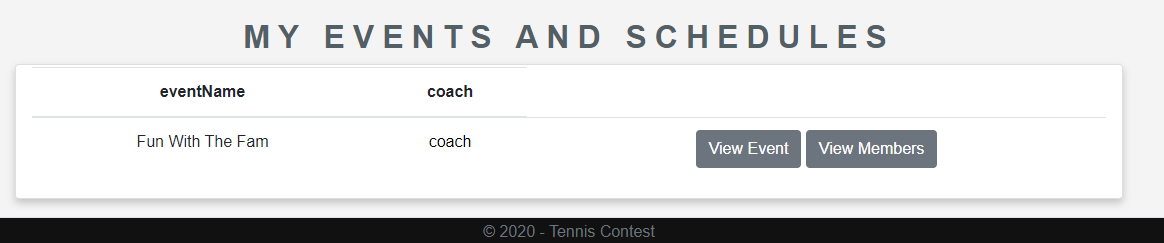
Coach Profile



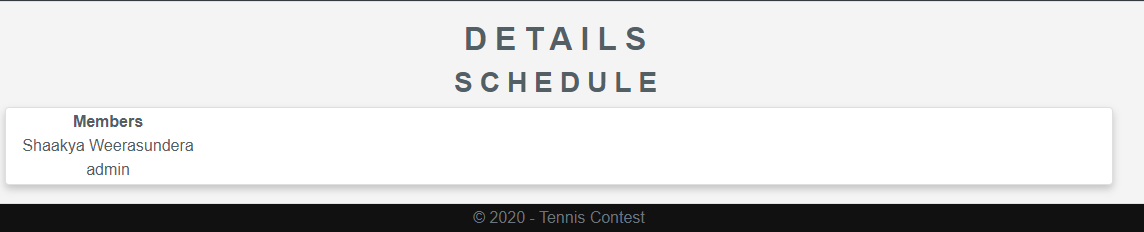
Edit



Schedules Page



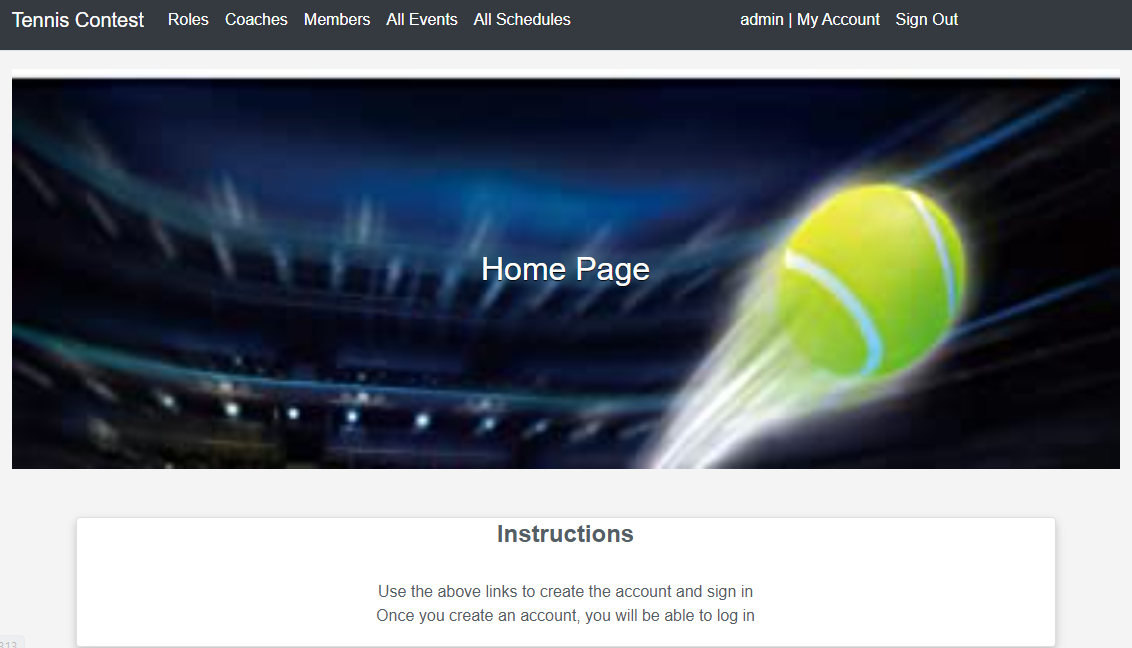
Coach can view current Members enrolled in the club page.



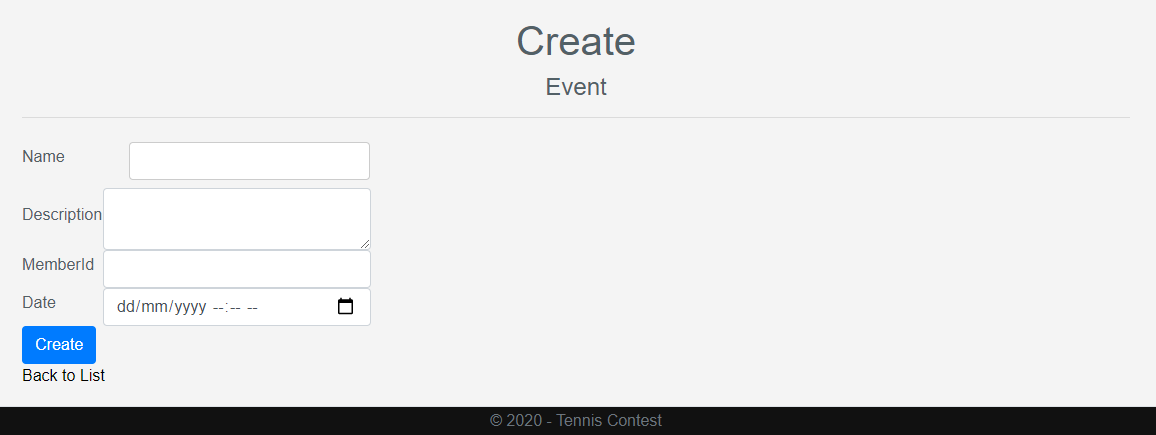
Admin Account:

* An admin user can create new Schedules and Events.
* Admin can also match the coach with the Schedule.
* In addition, the admin also can see all members in a list in Members page.
* Admin has privileges to view all the information in the club.

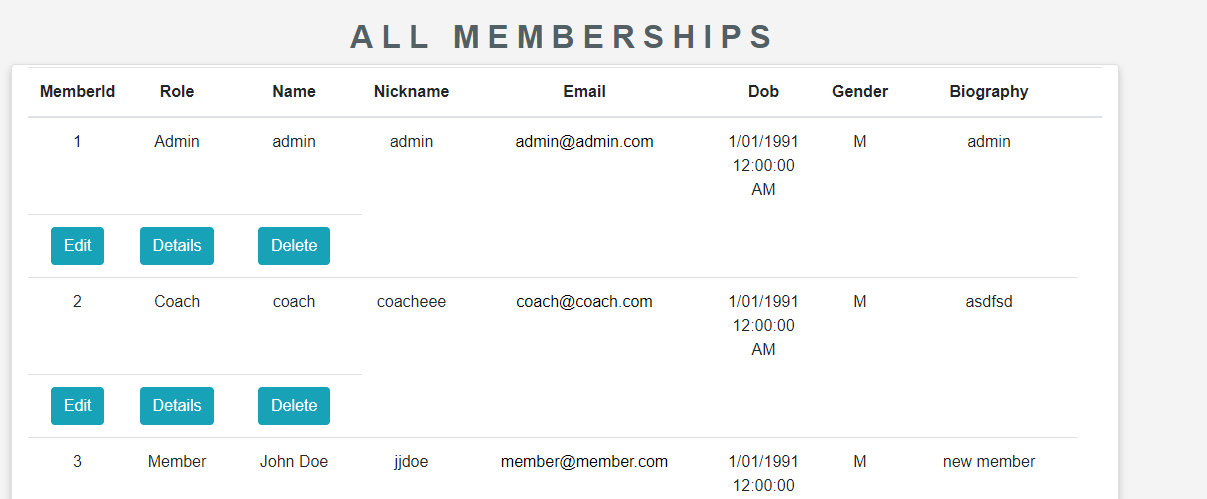
Home Page



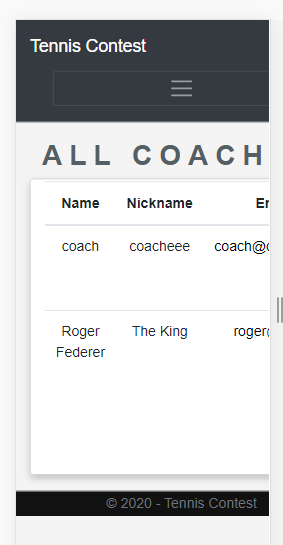
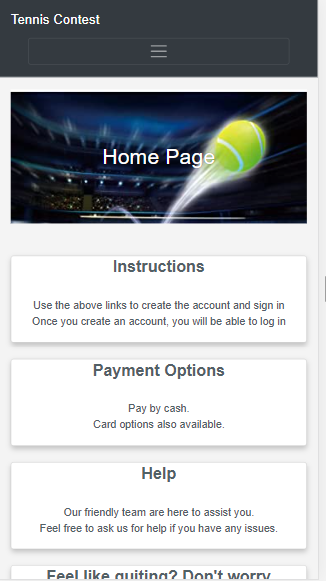
Create Event Page

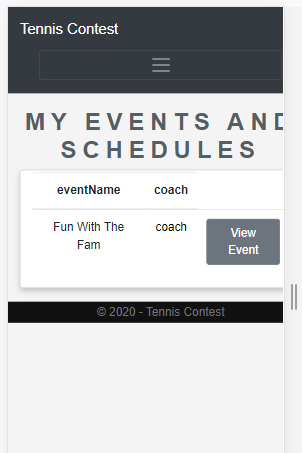


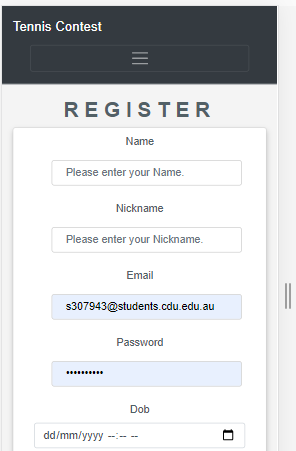
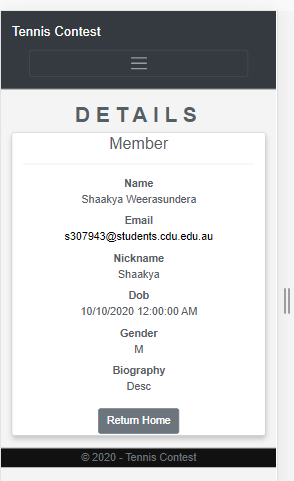
All Members Page

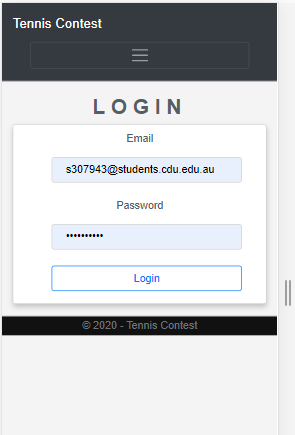


By following the W3Schools tutorial, the team applied CSS styles and made the app look nicer and more responsive. We used Chrome Development Tools to adjust the screen to a mobile size and see what the app looks like in a mobile device. The app looks nice and responsive after adding cards, as shown below:

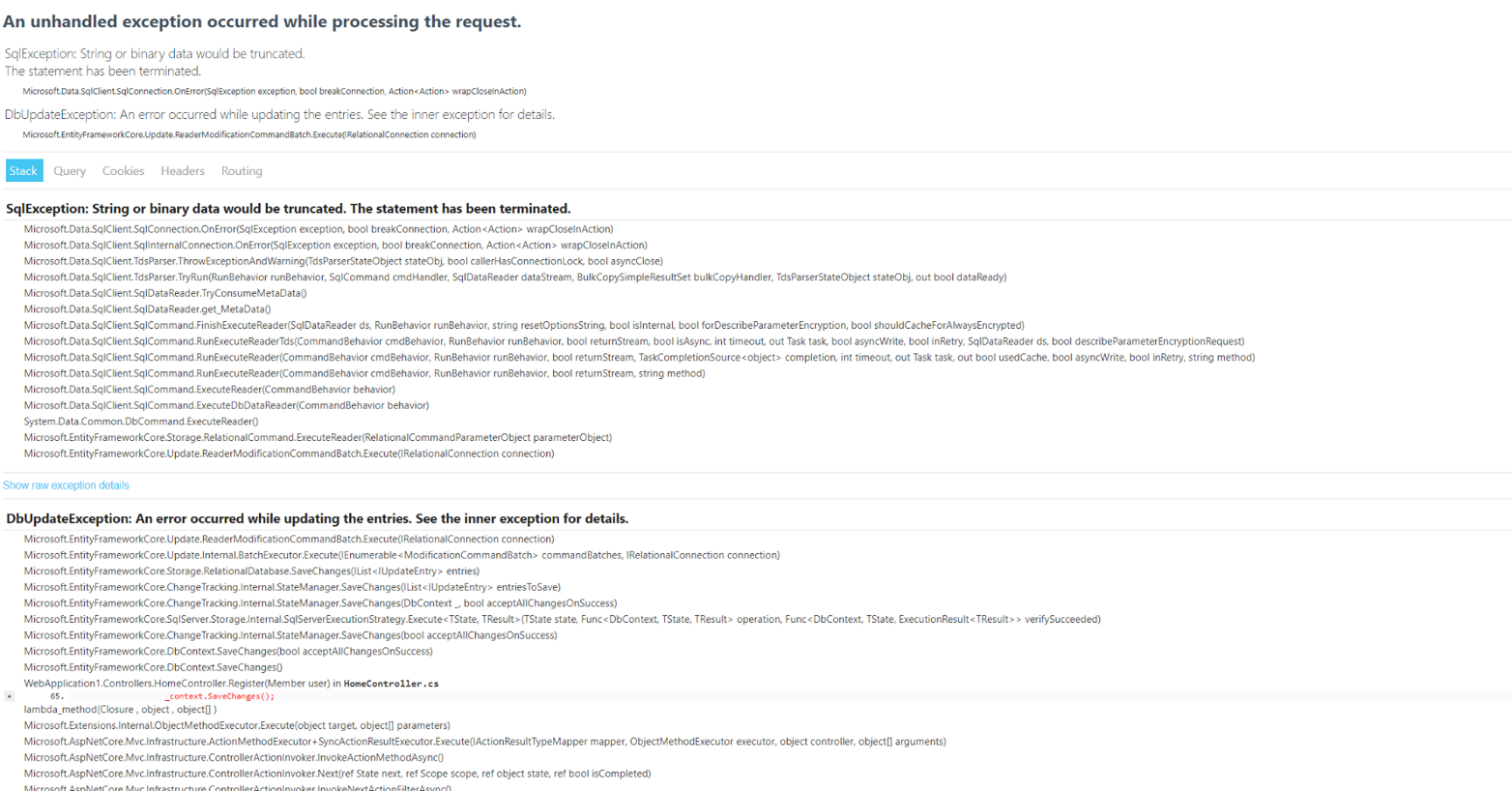








# Problems encountered and how they were solved:

When coding the web application, there were no doubt many errors that needed to be debugged in order to get the app functioning correctly. For instance, when a member registers for an account, the following **SqlException: String or Binary data will be truncated** occurred. The team discussed this together and found that the error was revolving around how the Entities **Schedules** and **Members were linked**. The Schedules was referencing the Members, and the Members was referencing the Schedules. The database was deleted, and migrations were made again. After this, the error was resolved. There was also an error when the user enters data in the Gender column with Male and Female. This was due to the length of characters in the database (it was varchar(1)). The length was readjusted and it works fine. From solving the errors, the team understood that it is important to ensure the data entered is within the range of the size of characters in the database. In addition, it is also important to understand the concept of foreign key relationships, when making relationships between entities. 

# References

Anderson, R. & Dykstra, T., 2020. *Part 4, add a model to an ASP.NET Core MVC app.* [Online]   
Available at: https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/adding-model?view=aspnetcore-3.1&tabs=visual-studio  
[Accessed 10 October 2020].

Microsoft, 2019. *Tutorial: Get started with EF Core in an ASP.NET MVC web app.* [Online]   
Available at: https://docs.microsoft.com/en-us/aspnet/core/data/ef-mvc/intro?view=aspnetcore-3.1  
[Accessed 10 October 2020].

w3schools.com, 2020. *How TO - Cards.* [Online]   
Available at: https://www.w3schools.com/howto/howto\_css\_cards.asp  
[Accessed 10 October 2020].