# Project Two Title: Socceringo

|  |  |
| --- | --- |
| **Name** | **Email Contact** |
| Abdul Khalil | Rehmank11@gmail.com |

## Logo

## **Icon Description automatically generated**

## Project Board Website Link and Screenshots

**https://github.com/absk786/Abdul-Project-three**

Graphical user interface

Description automatically generated

## Project Description (High Level):

This application will allow buys to securely sign into the application, browse tickets for the the clubs they wish to see play and then securely checkout and pay for the tickets.

### Motivation

One of the challenges with obtaining football tickets is that majority of the tickets are released as season tickets to fans and this means that throughout the the season there are very limited tickets to thousands of fans that do not get season tickets from the initial raffle. This leads to lots of highly inflated ticket prices and transactions which are very insecure and usually leaves fans with bad experiences

## User Story

**### User Story:**

```

AS a avid football fan,

I WANT to easily and safely be able to browse football tickets for the teams i like

SO THAT I can add items to my cart for checkout.

```

**### Acceptance Test for User Story:**

```

GIVEN an e-commerce website,

WHEN I open the site,

THEN I am able to see the app’s name, logo, and content.

WHEN I click on “Create an Account”,

THEN I am directed to creating a new account which will later be saved.

IF I click the “Sign In’ button and did not previously make an account,

THEN I will not be able to access any account.

IF I do not enter a valid username or password,

THEN I will also be unable to access any account.

WHEN I click on the “Sign In” button and do already have an account,

THEN I can sign into a previously made account and view saved items.

IF I chose to browse the website without creating an account,

THEN I can continue to browse and add items to my cart without creating an

account.

BUT I will not be able to proceed to the checkout unless I make an account.

WHEN I go through the application’s inventory,

THEN I have the ability to view items based on genre or trending selections

and see available records for purchase.

WHEN I click on a club,

THEN I am given the option to view the tickets for the upcomming matches for the club

WHEN I click on a match,

THEN I will be shown the available tickets in different areas for that matches.

WHEN I select on my cart,

THEN I am able to view the items I have saved for checkout and proceed

with the order.

WHEN I go through my cart,

THEN I am able to see my saved items in a list and a total price for all items.

IF I have not placed any items in my cart,

THEN if I click on my cart, it will tell me that it is empty.

WHEN I navigate through the application on a tablet or smartphone,

THEN the application is still responsive, and I can still view and save records

to my cart.

## Description

## APIs to be Used:

{list the Names, descriptions and URLs of the APIs being used for your project below, along with the reasons on why you are using it }

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Resource | URL | HTTP Verb (GET, POST) | Action | Used For |
| GraphQL |  |  |  |  |
| JWT |  |  |  |  |
| Stripe |  |  |  |  |

## Libraries to be Used:

{List client-side CSS and JavaScript Libraries being used in this project. List the names, descriptions and URLs for these libraries, along with the reason to use it . Examples include JQuery, Materials UI}

## Packages Required:

{Not Applicable for Project One}

## Minimum Viable Product (MVP) Requirements

{Describe here your minimum features that are required to make your project functional and presentable}

## Stretch Goals

* Select match specific tickets
* Interactive stadium layout
* Private sellers to create account and sell tickets online
* Add Hotel and travel packages with tickets

## Breakdown of Tasks (Ownership by Group Member):

Documentation – Abdul Khalil

Design - Abdul Khalil

Backend – Abdul Khalil

Frontend - Abdul Khalil

Testing - Abdul Khalil

## Schema

Example:

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Properties / Fields | Type | Details |
|  |  |  |  |
| User | firstName | String |  |
|  | lastName | String |  |
|  | email | String |  |
|  | password | String (scrambled) |  |
|  |  |  |  |
| UserList | User (many) | Array (list) |  |
|  |  |  |  |
| Product | Name | String |  |
|  | Description | String |  |
|  | Price | Number |  |
|  | SKU | Number |  |
|  | ID | String |  |

Etc.

## Associations:

Example:

* A userList has many users
* ProductCategories has a list of individual Product Category information
* A Product Category has one or more products

Etc.

## Validations:

{Describe what forms will require validations, what fields within a form are being validated, and what type of validation is required}

Example:

New User Form

* Required Fields
  + First Name (text)
  + Last Name (text)
  + Email (valid email address)
  + Password (minimum, maximum, character type)

Search on Map Form

* Required Fields
  + City (text)
  + Province / State / District (text)
  + Country (text)

Etc.

## Migrations and Seed Information

{Describe here any initial data that you might need to initially apply (populate) before your website will work. For example , a list of “default users” }

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| User | firstName | lastName | email | password | status |
|  | John | Doe | [john@doe.com](mailto:john@doe.com) | P@$$w0rd! | Active |
|  | Sally | Smith | [sally@smith.com](mailto:sally@smith.com) | T1geR\_1IlY | Inactive |
| … | … | … | … | … | … |

## Wireframes:

**A picture containing icon

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

## Graphical user interface Description automatically generated

## Graphical user interface Description automatically generated

## Logo, company name Description automatically generated

## Graphical user interface, application Description automatically generated

## Graphical user interface, application Description automatically generated

## Website “Components” and/or “Sections”

{ Based on your wireframes, describe the common components on the website “navigation”, “forms”, “headers”, “footers” either with images / and or content and describe the intention / usage of the components. In addition, describe the primary “views” or “sections” that your project will have, for example “About Us”, “Products and Services”, etc.

}

## Git Workflow, Website Directory Structure:

{describe your workflow and methodology that all team members will abide by when using Git, AND describe in text or imagery the directories and filenames and their descriptions that will be used for the project}

Example:

### GIT Workflow

Production Branch Name: Main

Development Branch Name: develop

All developers will create branches off develop according to the naming convention:

{

example:

***feature/lastname/ticketnumber/task i.e. feature/Apostol/001/navigation-bar***

***docs/lastname/ticketnumber/task i.e. docs/Wayne/020/update-README***

***bug/lastname/ticketnumber/bug-desc i.e. bug/Kent/030/fix-navigation-bar-color***

}

### Directory Naming and Structure; Filenames and descriptions

Only list the important files and folders.

|  |  |  |  |
| --- | --- | --- | --- |
| Directories | name | description | detail |
|  | images | Stores images |  |
|  | images/home | Images in home page |  |
|  | images/products | Product images |  |
|  | images/bio | Images for bio |  |
|  | css | Stores css | Main.css is principal stylesheet |
|  | css/nav | Navigation style |  |
| … | … | … | … |
|  |  |  |  |
| Files |  |  |  |
|  | Index.html | home |  |
|  | Contactus.html | Contact us page |  |
| … | … | … | … |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |