BetEx - A dynamic betting exchange web application

Colm Woodlock

G00341460

B.Sc.(Hons) in Software Development

April 26, 2019

Final Year Project

Advised by: Mr Daniel Cregg

Department of Computer Science and Applied Physics
Galway-Mayo Institute of Technology (GMIT)

Contents

Abstract

Abstract For my final year project, I was looking to create a project that would offer the chance for users to bet and lay on sporting events. A betting exchange is something I have become very interested in recent times and thought that this would be a great opportunity to research the subject and develop my own solution. I wanted to develop this as a web application where a user can create an account, top-up their account, place bets and also lay bets. The bets are populated and set via pulling data from other popular betting companies to get the best price possible. I created this application with a client and server pulling data from the data from the databases. It is a three-tier architecture application using a MySQL data base in the data tier, PHP on the logic tier and HTML and BootStrap for the Presentation tier.

Authors Colm Woodlock

Chapter 1

Introduction

When choosing my project, I wanted to pick something that was relevant to what I was interested in and something that I might find useful. This project would also have to highlight my existing skills but also offered some space to learn new skills and research a topic I may not be fully familiar with. With this criteria in mind I started to brain storm ideas on some of the things that I might like to do.

In the end I settled on a web application called BetEx. BetEx would be a betting exchange. My reason for choosing this is as a few years ago like many other college student was looking for some way to make some extra money on the side with as little effort as possible. That is when I stumbled across a site called OddsMonkey [1], on this site they claimed if you followed their guides and tips a user could claim free money that bookmakers were simply giving away. Having seen the offers for "Bet €20 get €20" all over television advertisements, billboards, newspapers and websites, I always thought to myself there must be a way of taking advantage of this but never gave it much more thought. Upon reading their guide I gave it a go and low and behold I had turned my free €20 bet into €15 of take home money with minimal effort. From there I did more research discovering new methods, all the while reaping the benefits of them. This has funded both my education and various holidays. A key component to matched betting is the betting exchange and not many exist on a large scale BetFair, Smarkets and Matchbook being the largest of them [2]. This lead me to the idea to develop my own platform to try and understand the workings behind the scenes of a betting exchange. I felt like there was an opportunity to make an exchange that had a much more user friendly interface as I found working with some of the established companies websites to be very confusing at the start. Once the project idea was decided on I then had to think about what technologies I wanted to use. I wanted to use many different types of technologies in order to demonstrate my

ability but also some that I would have to learn to show my capability to learn new things and incorporate them into an application. In the end I settled on a three tier structure containing a HTML and Boostrap presentation tier, a PHP logic tier which would offer new learning experiences and also for the data tier I would be using a MySQL database. From all of this I hoped that I would achieve a product in the end that would meet my expectation of both myself and the expectations worthy of a final year project in a Software Development course.

Chapter 2

Context

The context of my application is to offer a service to users who are new to laying bets. On the site a user can create an account, top-up their account and place a bet/lay from a list of real and regularly updated events.

- Log-in.
- Top-up.
- Place bet/lay.

2.1 Objectives

2.1.1 Home Page

The objective of the home page is to offer a landing platform for the user. Here they will be presented with the navigation for the website such as access to log in and register pages. A user who is not logged in may not have access to some other features presented on the home page.

2.1.2 Register Page

The register page offers a form to the user where they can securely create an account for the web application. Once the form is filled out correctly and successfully the user automatically sent out an email for confirmation.

2.1.3 Log in Page

The log in page allows users who have already registered for the site to access their account. Here they enter their email and their password which is stored

as a hashed value. The logic behind the page then compares the password entered to the hashed value that is stored in the database. If the passwords match the user is logged in and they now have access to more features.

2.1.4 Activation Page

The activation page is used for when the user registering. A code will be automatically generated and emailed to the user to their provided email account and once entered the user will then be fully registered.

2.1.5 Forgot Password Page

Similar to the Activation page the user will be sent a code that is automatically generated and emailed to their provided email account. Once the user enters this code and it is correct they will then have the opportunity to change their password.

2.1.6 Top up Page

Provided the user is logged in they will have access to the top-up page. Here the user will be able to enter their details and the top-up amount will be added to their account to be used when betting/ laying on events.

2.1.7 User Page

The user page will be available once the user is logged in, here it will display information that is personal to that user such as their active bets or their account credits.

2.1.8 Available Bets Page

On the available bets page the user will be able to see the events that are currently available to bet/lay on.

2.1.9 Bet/Lay Page

On this page, once the user has selected an event to bet on they will be given the opportunity to select an amount they wish to place a bet/lay on the event. Once it is placed it will be assigned to their account which they will be able to see on their user page.

2.2 Project Links

The full project is available here on GitHub:
https://github.com/cwoodlock/4thYearSoftwareProject

2.3 Chapter Review

This paper is broken down into various chapters ranging from the planning of the project to the design and implementation of a solution. The following sub-sections will give you a brief overview of each chapter in the paper.

2.3.1 Methodology

In this chapter I will discuss some of the methodologies I used during the various stages of creating the betting exchange web application. In this section I discuss Agile, version control, testing and sprints.

2.3.2 Technology Review

In the technology review chapter I review the various technologies that I used in the development of my project. I review everything from the back and front end technologies to development tools and version control.

2.3.3 System Design

In this chapter I give an explanation of the overall architecture of the betting exchange. I explain why I decided to do certain things and how I implemented those decisions into the project.

2.3.4 System Evaluation

In this chapter I evaluate the system in regards to robustness, testing, test results and limitations.

2.3.5 Conclusion

Finally in the conclusion I summaries the project against what I set out to do in the goals and objectives. I review the application from the various stages of development and I talk about the possible future developments with the project.

Chapter 3

Methodology

In this chapter i will discuss the methodologies used in this project. A methodology is just a way to plan and control the development process of a piece of software effectively. There are many different types of methodologies such as extreme programming or waterfall but for this project I tried to used Agile as my main methodology.

3.1 Agile Development

During the life cycle of this project I attempted to use agile as my main methodology whenever I could. I used this approach during all stages which included the research, design and implementation. In the initial stages of the project I was considering what methodology to use but ultimately decided on using agile as it is what is used by a lot of companies at the moment and I was interested in trying to replicate their approach to developing software.

There are different styles of agile development such as KanBan or extreme programming, but the one I chose for my approach to the project was scrum. Scrum is an agile methodology in which a cycle is carried out in what are known as sprints.

Throughout the life cycle of the project I held weekly meetings with my project advisor. In these meetings we would discuss what I had done in the previous week and what I was planning to achieve in the following week. Whatever was decided at these meetings, I then created cards on the GitHub project board on the projects page. This helped me focus on the task at hand and I found using this project board to be very useful in the development of this project. Once I started the particular task I would edit the project board and move the card to the in progress column and upon completion of the task move it to the completed section.

3.2 Version Control

Throughout the life cycle of the project I used GitHub for the version control. GitHub is a hosting service for version control. What I did was created a repository on GitHub and during the development of the project I would commit whatever progress I had made during that session to GitHub to record my progress.

Tracking the source code I was committing to GitHub was not the only service I used, I also used the task management tool. This allowed me to create tasks that I wanted to complete and assign them to various sections such as to-do or in progress.

GitHub also provided an opportunity for me to share my work with my project advisor in weeks I could not attend the project meeting. On the repository you can see when new things are committed and what exactly was updated. Another feature I used with GitHub was the ability to rollback the project if needed. Github Tracked the entire progress of the project and at any time I could revert back to any of the previous commits and this proved very useful.

3.3 Testing

For my application I needed to decide how I was going to test the application as I could not use something such as J-unit. In the end I decided to go with white and black box testing. We had looked at this method of testing in a previous module and I thought it was fit the purposes of testing that I needed.

3.3.1 White Box Testing

In white box testing the tester can see in internal workings of the software. White box testers have a full and comprehensive knowledge of the internal makeup of the software and are usually software developers themselves. This is a role I thought I suited and so I carried out the software tests. What I did was I wrote out some features I would like to test and the expected outcome. For example if I was testing the log-in system, the expected outcome would be that the user is logged in and is presented their user page. I would then test this and follow the path the program follows as it moves down through the code.

3.3.2 Black Box Testing

Black box testing is a way of testing a piece of software without knowing the internal functionality of the software being tested. A black box tester has no knowledge of the internal design, structure or implementation of the software and are often not software developers. For this I asked some of my friends to test out certain features of the software. The web application was hosted on the Google cloud platform and so I was able to send them a link and ask them to try and perform certain functions. Again I would have made a list of expected outcomes and asked them to recommend any changes if they found any problems themselves. I found this very useful as I often skipped over something small from being so used to looking at it the whole time.

3.4 Sprints

In this section I will be talking about sprints. Each of the sprints I tried to complete in a timely manner and as effectively as possible. Each sprint represents a vital part of the application.

3.4.1 Sprint 1 Research

This was my first sprint and I used it to research existing exchanges. I focused on understanding what was user-friendly and how I could potentially improve on existing solutions.

3.4.2 Sprint 2 Design

During this sprint I focused on the design of the application. Not the aesthetic design but rather the database structure and web application structure.

3.4.3 Sprint 3 Database Connection

Once the design was complete it was time to start the first of the implementation sprints. For this sprint the objective was to set up a basic three-tier solution. For this I stored some data in a database and using php I presented it on a web page.

3.4.4 Sprint 4 Log-in and Registration

Once the basic pipeline was implemented I then decided to work on a log-in system. This log-in system would be pretty comprehensive in regards to

the validation checks, cleaning the data before storage in the database and creating a unique session for each user.

3.4.5 Sprint 5 Website Design

Once the log in system and registration system was implemented I then decided to work on the general website design. I created a home page where a user could log in from and a user page available only to the logged in user and also general website navigation.

3.4.6 Sprint 6 User Validation

This was the sprint where I finished off the user validation I added a system to email the user an activation code when registering and also an activation code when trying to reset passwords. All of these help with the security of the site.

3.4.7 Sprint 7 Sessions

During this Sprint another feature I added in this, was the remember me function. This used session data to remember the user from a particular session and not require the user to log in providing the session has not ended.

3.4.8 Sprint 8 Top up account

During this sprint I wanted to add functionality to the user to be able to top up their account have have credit to use on betting/laying events.

3.4.9 Sprint 9 Display Data from Database

This was a very important sprint as it was the one where I pulled the data on the events and presented them to the user so that they could then place a bet or lay on the event.

3.4.10 Sprint 10 Place a Bet/Lay

Now that the data was being displayed, during this sprint I allowed the user to place bets or lays on their selected event. This bet/lay would then be assigned to their account so that the user could see what bets/lays they have placed.

3.4.11 Sprint 11 Python Scraper

The final sprint was to implement a python scraper that would be used to populate the database for the events. This script would be running on the web server and continually updating the data that was being presented to the user.

Chapter 4

Technology Review

In this section, I will cover the overall design and architecture of the application. I will do this through using code snippets to help to understand the design. The technology review chapter will be broken up into three sub sections which will be the data tier, logic tier and presentation tier. An additional subsection will be used for other technology I used that may not fit into the 3 tier structure.

4.1 Data Tier

The data tier represents the overall stored data and gives the ability to store, access, update and delete certain data. For my application I have only one database. This database is used to store user information, events and betting/laying information.

4.1.1 MySQL

I chose MySQL to handle all of my database needs as I have used it before and I think it works very easily with php. There are some issue with using MySQL such as sql-injection but I have made some simple functions to mitigate this in my application. In this database I am storing table which will contain all the user, event and placed bets data. During the making of the application these tables and the structure of the database changed many times using a MySQL database with phpMyAdmin made it very easy to change and rearrange the database when needed.

As of writing this dissertation I do not have it implemented but the original plan was to get a working web scraper in python and scrape data into a database. Scraping data to a sql file isn't too challenging and this was

another reason that I chose to use a MySQL database. Overall it fulfilled my needs and the needs of the application.

4.2 Logic Tier

4.2.1 Php

PHP is a general-purpose programming language originally designed for web development in 1994 by Rasmus Lerdorf [3] . There are a few reasons I chose to use PHP in my project over more modern and appealing node js and go, one of these reasons was we covered javascript and go on this course in some modules and I wanted to show my ability to learn something different that we havn't been though on this course. Another reason being is that I had some experience with PHP before on a a previous project I did and I enjoyed PHP development. So with these in mind I thought it suited what I wanted to achieve in this application.

4.2.2 Python

Python is a programming language that is very interesting and very powerful. Having had some experience with Python throughout the course I wanted to incorporate it somehow into my project. The plan was to use the many APIs available within python to scrape data from a website then use it to create a sql database of information that would be continually updated. As I have mentioned before I have not implemented this into the project in the time of writing this but I hope to have it implemented in the live project, available on GitHub.

4.3 Presentation Tier

4.3.1 HTML

HTML5 or Hypertext Markup Language is a standard markup language for creating web pages and web applications. Using it in conjunction with PHP was an integral part of my project. Using it allowed me to prototype pages quickly to check interactions and functions I had written in PHP. Once the page demonstrated that it was functioning correctly it was very simple to implement Bootstrap into the pages to make them appear more aesthetically pleasing.

4.3.2 jQuery

jQuery is a javascript library. Using this allowed me to add animations to certain interactions. Combined with the styling from Bootstrap some of this animations made the functionality of the web application not only interactive but also aesthetically appealing.

4.4 Additional

4.4.1 CSS

I used CSS or Cascading Style Sheets in a number of places in this project. CSS allows the user to style their web page from the standard HTML page. This adds uniqueness to the application and also appeals to the eye. The main place I used this was in editing the existing Bootstrap.css file that is provided when using Bootstrap. I edited it to provide a custom Bootstrap theme that would be unique to my web application.

4.4.2 Bootstrap

There are many reasons to use Bootstrap and having used it before it was an easy decision for me ti include it in my project. Bootstrap is a framework for HTML, CSS and JS for responsive development. One of the main reasons I decided on using it is that it saves time when you are designing your web pages. It is very easy to use templates and classes and simply plug them into your application and in most cases this will work almost immediately. Another reason I decided to used Bootstrap was that it is very easy to customise. Making this themes unique and personal to me is very easy to do and this was again another factor into why I chose to use Bootstrap. The is also extensive documentation that comes with Bootstrap, providing live examples and all the various options available to the user, this makes it very easy to decide on what you want to implement into your web application and then do it.

4.4.3 LaTeX

LaTex is a typing system that is used throughout industries. It works on a scripting programming language. LaTex is not as simple and as user friendly as something such as Microsoft Office or Google docs however it is a much more powerful tool in regards to customisation and referencing in particular. I have used LaTex in previous modules and has since become my regular

document writing application. For instance this dissertation is being written in LaTex. It can be quite difficult to understand in the beginning however with some practise and understanding of some of the easier commands you quickly get the hang of it. Another feature from LaTex is the ability to control the layout of the file, using simple commands a user can change the appearance of a file very easily.

4.5 Technologies Used

4.5.1 Sublime Text

As I was developing the majority of my project in PHP I did not require a large IDE such as Microsoft Visual Studio, instead I could use a simple light weight editor. I searched for some alternatives and found Sublime Text was the application that suited my goals best. It allowed for easy project management, readable colour scheme for php and also git integration that allowed for quick and easy commits to GitHub. Overall I have no complaints about Sublime Text and it accommodated everything that I asked for it in regards to the web application that I made.

4.5.2 phpMyAdmin

phpMyAdmin was used to manage the databases I was using throughout the development of the project. Having used it before it was very easy to view and manage databases. The UI may be slightly confusing to users who are not familiar with the layout however it doesn't take long to get used to it and navigate the application with ease.

4.5.3 GitHub

GitHub is a hosting service and it allows users to share and collaborate on projects together. It is often used in open source projects. Projects hosted on GitHub are called repositories and it is here that a user can save their projects to, pull down changes that a collaborator has made or push up changes they have made themselves. It is a very powerful tool and one I have become used to using with all of my projects.

Despite not collaboration with anyone on this project Github still proved a valuable tool. One of the reasons it proved a valuable tool was that it allowed me to track my progress and also allowed me to share my progress with my project advisor though out the life cycle of the project.

4.5.4 Google Cloud Services

Google Cloud Service provided a a quick and easy solution to host a live version of my web app. Having installed wamp onto the virtual machine it was very simple to configure and display a static version of my web application at a certain ip address.

4.5.5 WAMP

WAMP stands for Windows Apache MySQL PHP and it is a package used by PHP developers to develop web apps. This was the perfect too for me as it comes installed with Apache server , MySQL database , phpMyAdmin and PHP all things I had identified that I wanted to use in my project. having used it before as part of a module it proved an ideal candidate to include into my project and worked without any issues.

4.5.6 Overleaf

Overleaf is an online real time editor used for editing LaTex. This means that I did not have to install specialist software in order to edit the LaTex files that I would be using to write my dissertation for this project. I had used it before in another module and was familiar with the user interface.

Overleaf offers a lot of features that was suitable for my project. I could share my pr ogres with my project advisor and it also offers the opportunity to link the files directly to your GitHub repository.

4.5.7 Cmder

Commander is a console emulator for Windows that allows the user to use some Linux commands on a windows machine. I mainly used this for merge conflicts and other issues I was having with GitHub when pulling or pushing commits. This wasn't too often but it was a tool that I found very useful and worth mentioning in this project.

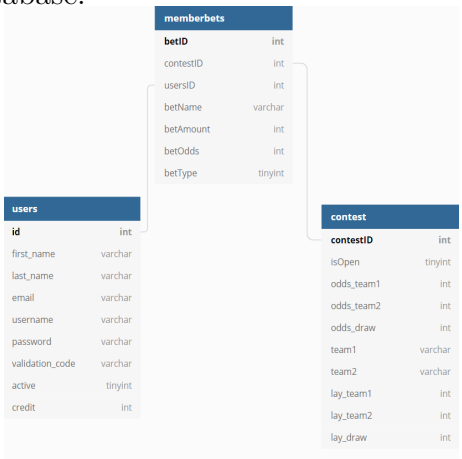
Chapter 5

System Design

In this section, I will be going over the overall design and architecture of the application. I will go though some code snippets that I will use as a visual aid to help you get a better understanding of what is going on in the application. I have broken down this section into 3 parts, the data tier, logic tier and the presentation tier.

5.1 Data Tier

The data tier represents the overall stored data in the application and gives the user the ability to store, read, update and delete the data contained within. I have a single database that is used to store the users, the events and also the bet information. The figure below shows the structure of the database.



5.1.1 MySQL

I chose MySQL to handle off the database requirements of the application as I was familiar with it and there are many resources online. I wanted to create a log in system that allowed a user to register, log in if they already have an account, account verification, forgotten password functionality and remember me functionality.

id	first_name	last_name	email	username	password	validation_code	active	credit
7	Brian	O'Connor	boconnor@gmail.com	boconnor	202cb962ac59075b964b07152d234b70	723e1ed8f2bf00f4de70ad8728645696	1	150

In the image above you can see I am storing the password and validation code as hashed values. This is achieved in the functions during the registration phase.

```
$password = md5($password); //hash password

//Hash password to create validtion code
$validation_code = md5($username + microtime());
```

Overall phpMyAdmin in conjunction with the mySQL database I created was very easy to work with and provided all the functionality I wanted to implement.

5.2 Logic Tier

The logic tier refers to the logic behind your application. This is where the majority of the data manipulation is done. This can include creation of new data, storing data, updating and deleting data. To handle this logic I decided to use PHP. It was very easy to integrate the database into the php logic.

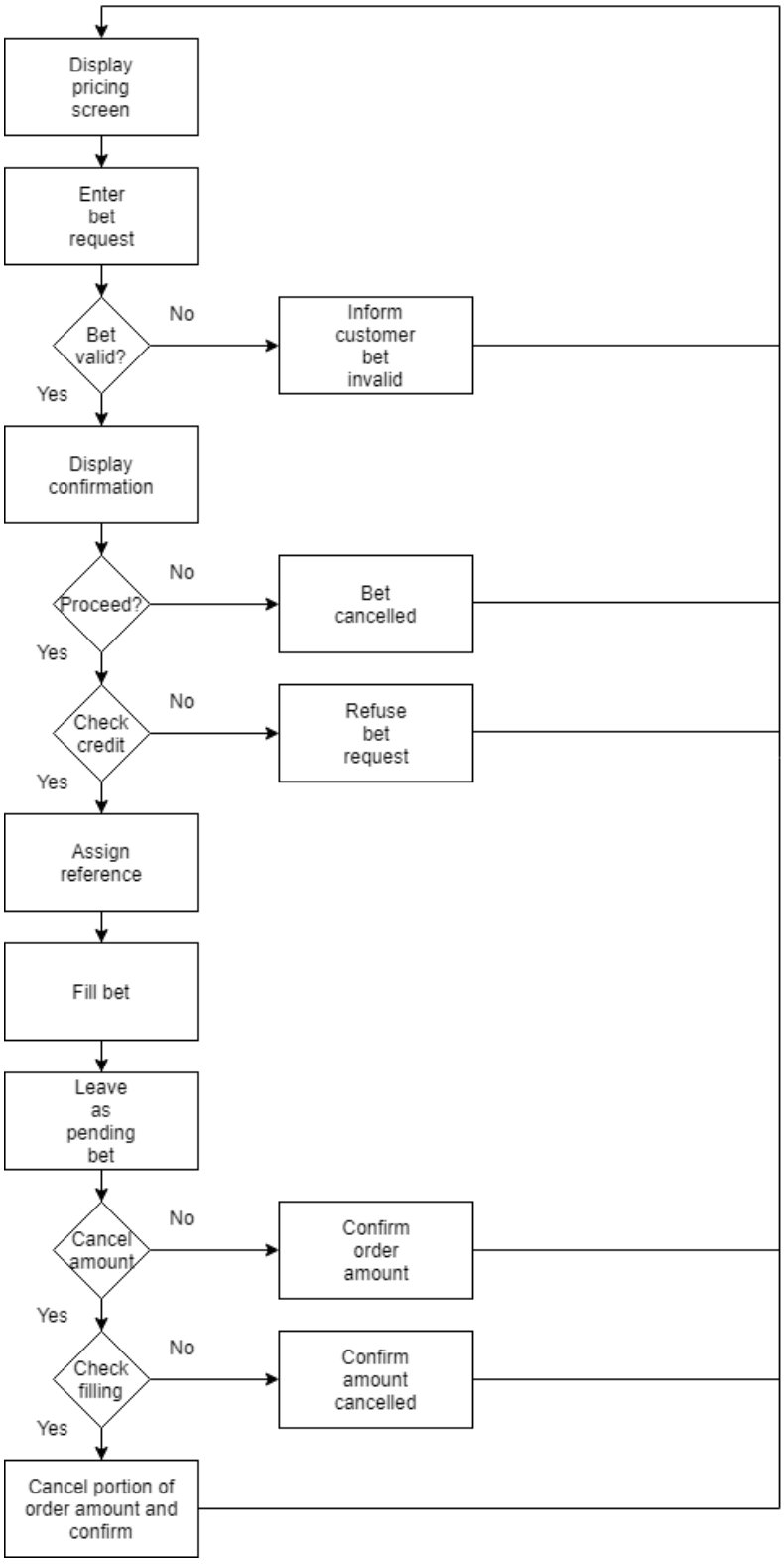
The first thing I wanted to establish in the data tier was a solid connection that I could all on each time I wanted to access the database, I also wanted a few key functions that I would use repeatedly throughout the project. Below is an image of these functions which I placed in their own php file called db. This php file would be included in every php file I wrote after this.

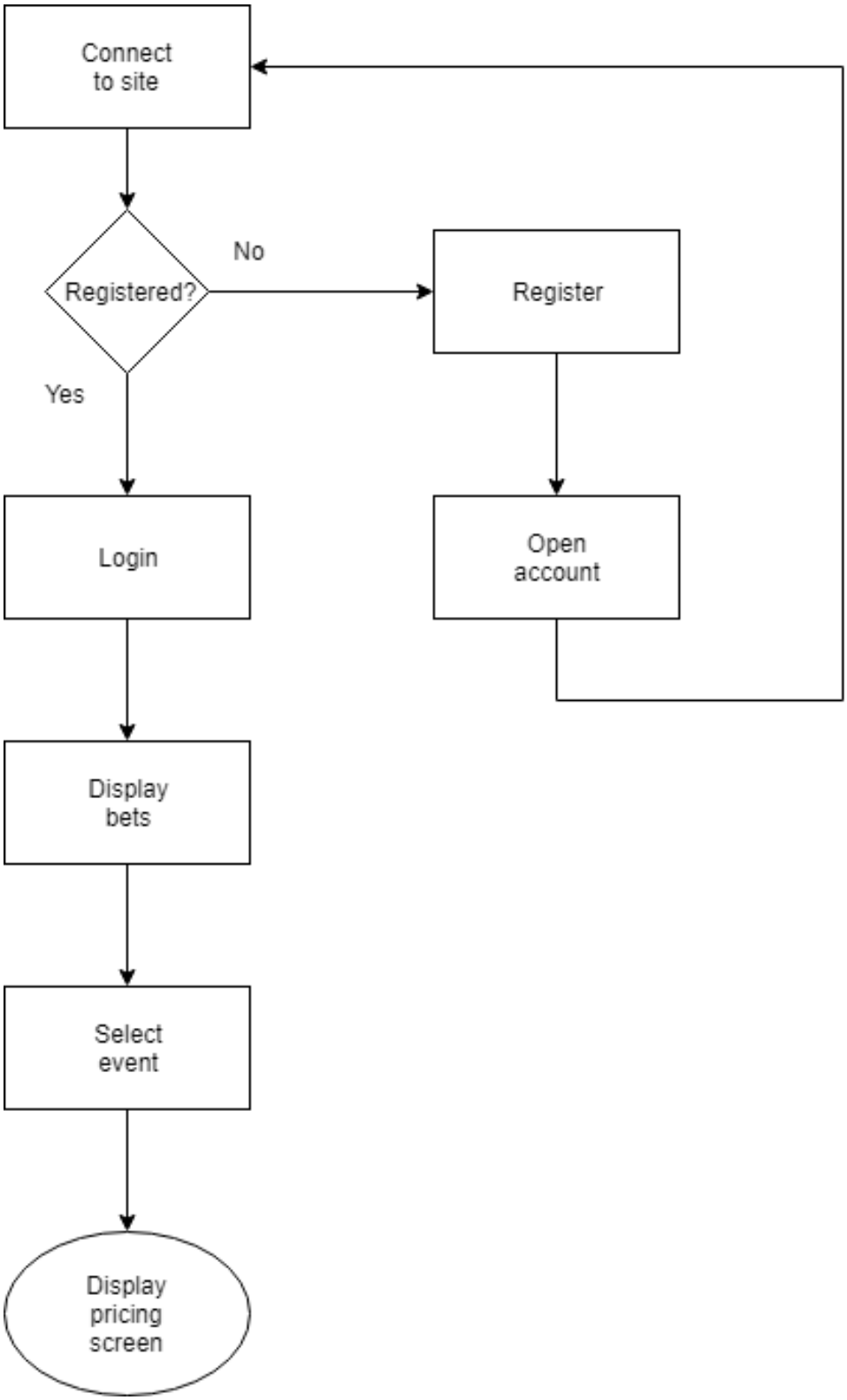
db.php

```php
<?php

$conn = mysqli_connect('localhost', 'root', '', 'login_db');


function escape($string){ //This will escape the data

    global $conn;

    return mysqli_real_escape_string($conn, $string);
}

function query($query){ //Function to help with querying

    global $conn;

    return mysqli_query($conn, $query);
}

function confirm($result){ //Function to ensure data is ok

    global $conn;

    if(!$result){
        die("Query Failed" . mysqli_error($conn));
    }

}

function fetch_array($result){ //retrieve the data

    global $conn;

    return mysqli_fetch_array($result);

}

function row_count($result){ //Count the rows in the database

    return mysqli_num_rows($result);

}

?>
```

Most of these functions are pretty standard and are just ways to access existing functions easier however one I would like to highlight is the escape function. This is used to sanitise the data that is being passed into it this is to help minimise the risk of sql injection which is a common problem with a sql database.

Below are two UML diagrams that I tried to follow in regards to how the site would be used and how I would implement various functions.


Display pricing screen
Enter bet request
Bet valid?
No
Inform customer bet invalid
Yes
Display confirmation
Proceed?
No
Bet cancelled
Yes
Check credit
No
Refuse bet request
Yes
Assign reference
Fill bet
Leave as pending bet
Cancel amount
No
Confirm order amount
Yes
Check filling
No
Confirm amount cancelled
Yes
Cancel portion of order amount and confirm


Connect to site
Registered?
No
Register
Yes
Login
Open account
Display bets
Select event
Display pricing screen

The next thing that I felt was appropriate to implement was the log in system. The logic being this was simple enough as it has been done many times over and so I has able to use these examples as reference for my system. To create the register system there were some functions I thought were important to include was a check to see if the user name exists already in the database and also if the email already exists. Below is the code that I used to complete these checks.

```
function email_exists($email){ //checks if email was used before

    $sql = "SELECT id FROM users WHERE email = '$email'";

    $result = query($sql);

    if(row_count($result) == 1){
        return true;
    } else {
        return false;
    }

}

function username_exists($username){ //checks if username was used before

    $sql = "SELECT id FROM users WHERE username = '$username'";

    $result = query($sql);

    if(row_count($result) == 1){
        return true;
    } else {
        return false;
    }

}
```

Next I had to implement these checks into a validation system. This was my second layer of validation as there was some validation check already happening with HTML5.

functions.php

```php
function validate_user_registration(){ //Function will validate the user
    $errors = [];
    $min = 3;
    $max = 20;

    if($_SERVER['REQUEST_METHOD'] == "POST"){

        $first_name = clean($_POST['first_name']);
        $last_name = clean($_POST['last_name']);
        $username = clean($_POST['username']);
        $email = clean($_POST['email']);
        $password = clean($_POST['password']);
        $confirm_password = clean($_POST['confirm_password']);

        if(strlen($first_name) < 3){

            $errors[] = "Your first name cannot be less than {$min} characters";
        }

        if(strlen($last_name) < 3){

            $errors[] = "Your last name cannot be less than {$min} characters";
        }

        if(strlen($username) < 3){

            $errors[] = "Your username cannot be less than {$min} characters";
        }

        if(strlen($first_name) > 20){

            $errors[] = "Your first name cannot be more than {$max} characters";
        }

        if(strlen($last_name) > 20){

            $errors[] = "Your last name cannot be more than {$max} characters";
        }

        if(strlen($username) > 20){

            $errors[] = "Your username cannot be more than {$max} characters";
        }

        if($password !== $confirm_password) {
            $errors[] = "Your passwords do not match";
        }

        if(email_exists($email)){

            $errors[] = "{$email} is already registered";

        }

        if(username_exists($username)){

            $errors[] = "{$username} is already registered";

        }

        if(!empty($errors)){

            foreach ($errors as $error) {

                //Display validation error
                echo validation_error($error);
            }
        } else {

            if(register_user($first_name, $last_name, $username, $email, $password)){

                set_message("<p class='bg-success text-centre'> Please check your email or spam folder for activation link.</p>");

                //redirect to home page
                redirect("index.php");
```

Providing the user had passed all these checks the information would then be passed into the registeruser function where the users information would be stored in the database and a validation code would be generated which would then be sent out to the user via email. It is worth noting that this function does not work on localhost unless you configure the localhost to do so.

```php
function register_user($first_name, $last_name, $username, $email, $password){

    $first_name = escape($first_name);
    $last_name = escape($last_name);
    $username = escape($username);
    $email = escape($email);
    $password = escape($password);

    if(email_exists($email)){
        return false;
    } else if(username_exists($username)){
        return false;
    } else {

        $password = md5($password); //hash password

        //Hash password to create validtion code
        $validation_code = md5($username + microtime());

        $sql = "INSERT INTO users(first_name, last_name, username, email, password, validation_code, active)";
        $sql.="VALUES('$first_name','$last_name','$username','$email','$password','$validation_code', 0)"; //append query onto previous

        //Confirm the query
        $result = query($sql);
        confirm($result);

        $subject = "Activate Account";
        $message = "Please click the link below to activate your account:

        http://localhost/4thYearSoftwareProject/activate.php?email=$email&code=$validation_code";

        $header = "From: noreply@betex.com";

        send_email($email, $subject, $message, $header);

        return true;
    }

}
```

At this point the account is created however the account must be activated in order to use it. If an email is sent out the user can click the link and they will be sent to an activation page on the site where they can enter the validation code that was also sent out in the email. Below is the code that is used to activate the account of the user.

```php
function activate_user(){
    //Test URL http://localhost/4thYearSoftwareProject/activate.php?email=jsmith@gmail.com&code=441c10bb769d04ff03ee47c77ce6bd42
    if($_SERVER['REQUEST_METHOD'] == "GET"){

        if(isset($_GET['email'])) {

            //Get the variables for email and the validation code from the url
            $email = clean($_GET['email']);

            $validation_code = clean($_GET['code']);

            //SQL statement and confirm the validation
            $sql = "SELECT id FROM users WHERE email = '".escape($_GET['email'])."' AND validation_code = '".escape($_GET['code'])."' ";
            $result = query($sql);
            confirm($result);

            if(row_count($result) == 1) {
                //Set the user to the active state
                $sql2 = "UPDATE users SET active = 1, validation_code = 0 WHERE email = '".escape($email)."' AND validation_code = '".escape($validation_code)."' ";
                $result2 = query($sql2);
                confirm($result2);

                //Set the message in session
                set_message("<p class='bg-success'>Your account has been activated please login</p>");

                //Redirect the user to login
                redirect("login.php");


            } else {

                set_message("<p class='bg-danger'>Sorry your account could not be activated </p>");

                redirect("login.php");


            }

        }
    }
}
```

Now the user has a fully activated account and can now log in. Much like the register system the log in system must have similar validation. This is to ensure the correct data is being entered and no input fields are being left blank or contain invalid data.

```php
function validate_user_login(){ //Function will validate the user when loging in

    $errors = [];

    $min = 3;
    $max = 20;

    if($_SERVER['REQUEST_METHOD'] == "POST"){

        $email = clean($_POST['email']);
        $password = clean($_POST['password']);
        $remember = isset($_POST['password']); //Checks for the remember me check box

        if(empty($email)) {
            $errors[] = "Email field cannot be empty";
        }

        if(empty($password)) {
            $errors[] = "Password field cannot be empty";
        }


        if(!empty($errors)){

            foreach ($errors as $error) {

                //Display validation error
                echo validation_error($error);
            }
        } else {
            if(login_user($email, $password, $remember)){
                redirect("admin.php");
            } else {
                echo validation_error("Your information was not correct");
            }
        }

    }
}
```

Again once the user passes the validation their information is passed into a log in function to log the user in. In this function the hashed password is retrieved from the database and compared to the hashed version of the entered password. If there is a match I the application then starts a session and saves the logged in email into the session. This will be used at a later time when we discuss the remember me functionality.

```php
function login_user($email, $password, $remember){ //Function to log the user in by comparing what they enter to the database

    $sql = "SELECT password, id FROM users WHERE email = '".escape($email)."' AND active = 1";

    $result = query($sql);
    if(row_count($result) == 1){

        //Get the password from the database
        $row = fetch_array($result);
        $db_password = $row['password'];

        //de-crypt the hashed password from the database
        if(md5($password) == $db_password){

            if($remember == "on"){
                setcookie('email', $email, time() + 86400); //the cookie will expire after 1 day in secs
            }

            //Save email in the session
            $_SESSION['email'] = $email;


            return true;
        } else {
            return false;
        }

        return true;
    } else {
        return false;
    }

}
```

An important part of the web application is knowing if the user is logged in. This allows the application to allow logged in users access information and functionality that is otherwise unavailable to users who are not logged in. I use a simple function to check this.

```php
function logged_in(){ //This function will make sure that the user stays logged in when they log in

    if(isset($_SESSION['email']) || isset($_COOKIE['email'])){
        return true;
    } else {
        return false;
    }

}
```

The final piece of the log in functionality that I wanted to get working effectively was the recover password functionality. This was a similar function to the one I used when activating a users account. Again in this function there is an email sent out containing the validation code to recover the password.

```php
function recover_password(){ //his fucntion will recover the password

    if($_SERVER['REQUEST_METHOD'] == "POST"){ //post request

        if(isset($_SESSION['token']) && $_POST['token'] == $_SESSION['token']){

            $email = clean($_POST['email']);

            if(email_exists($email)){

                $validation_code = md5($email + microtime());

                //Cookie so that webpage isn't available all the time
                setcookie('temp_access_code', $validation_code, time()+ 60);

                $sql = "UPDATE users SET validation_code = '".escape($validation_code)."' WHERE email = '".escape($email)."' ";
                $result = query($sql);
                confirm($result);

                $subject = "Please reset your password";
                $message = " Here is your password reset code {$validation_code}

                Click here to reset ypur password http://localhost/4thYearSoftwareProject/code.php?email=$email&code=$validation_code
                ";
                $headers = "From: noreply@betex.com";

                if(!send_email($email, $subject, $message, $headers)){

                    echo validation_error("Email could not be sent ");

                }

                set_message("<p class='bg-success'>Please check your email or spam folder for a password reset code</p>");
                redirect("index.php");

            } else {
                echo validation_error("This email does not exist");
            }
        } else {
        //If the token isnt set redirect to index
        }


        if(isset($_POST['cancel_submit'])) {

            redirect("login.php");
        }
    }
}
```

Another function that I would like to highlight is the function used to place a bet. In this function I retrieve data from various tables and pass the relevant data into the table which will store the information on the stored bet. I use similar functions for when the user is laying a bet.

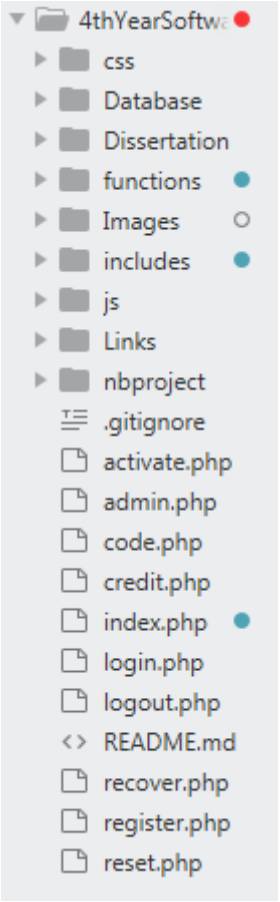
```php
//Place Bet on team 1
if(isset($_POST['placeBetTeam1'])){
        $sql = "SELECT contest.contestID, contest.team1, contest.odds_team1 FROM contest";
        $result = query($sql);

        confirm($result);
        $row = fetch_array($result);


        $sql1 = "SELECT users.id, users.email FROM users WHERE email = '".escape($_SESSION['email'])."'";
        $result1 = query($sql1);

        confirm($result1);
        $row1 = fetch_array($result1);

        $contestID  = $row['contestID'];
        $team1      = $row['team1'];
        $odds_team1 = $row['odds_team1'];

        $userID     = $row1['id'];
        $email      = $row1['email'];

        $amount = $_POST['Team1BetAmount'];
        $betType = '0';


        //echo $contestID, ' ' , $team1, ' ' , $odds_team1, ' ' , $userID, ' ' , $email;

        $sql3 =     "INSERT INTO memberBets
                    SET contestID= '".$contestID."',
                    usersID= '".$userID."',
                    betName= '".$team1."',
                    betAmount= '".$amount."',
                    betOdds= '".$odds_team1."',
                    betType= '".$betType."'
                ";
        $result3 = query($sql3);

        confirm($result3);

        redirect('admin.php');
```

5.3 Presentation Tier

In the presentation tier I will look at the website structure, individual pages and also deployment and hosting on a virtual machine. I will talk about design choices I made and the overall design of the application.

5.3.1 Website Structure

In this section I will talk about the structure of my project. Below is an image of the project as it appears in Sublime Text.



In this image you can see that all of the main pages that relate directly to the website structure are in the root. From here you can then see in the folder js. In this folder you can find all the javascript related files. In here we have the jquery.js and bootstrap.js files. Above this we have the includes folder. In this folder I have placed all the php files that will be used as includes on each of the main files. An example of this is the header and footer files are in this folder as well as the nav bar file.

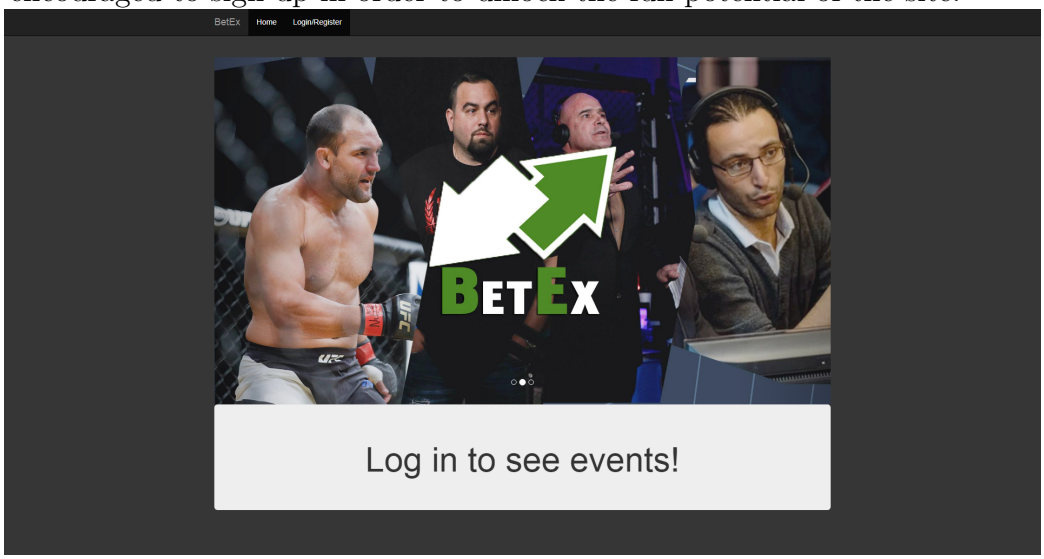
Above this are images that are being used in this dissertation for example. All images on the web application are hosted online. Above the images folder we have the functions folder. Contained within this folder we have init.php which contains a function call for obstart and sessionstart which are used in the application and must be called when the application start and it also includes db.php and function.php which are two files that are also contained within the functions folder. db.php contains all of the database connection and querying functions while functions.php is a file full of all the various functions that I use throughout the application.

Above that we have the database folder in her contains a copy of the sql

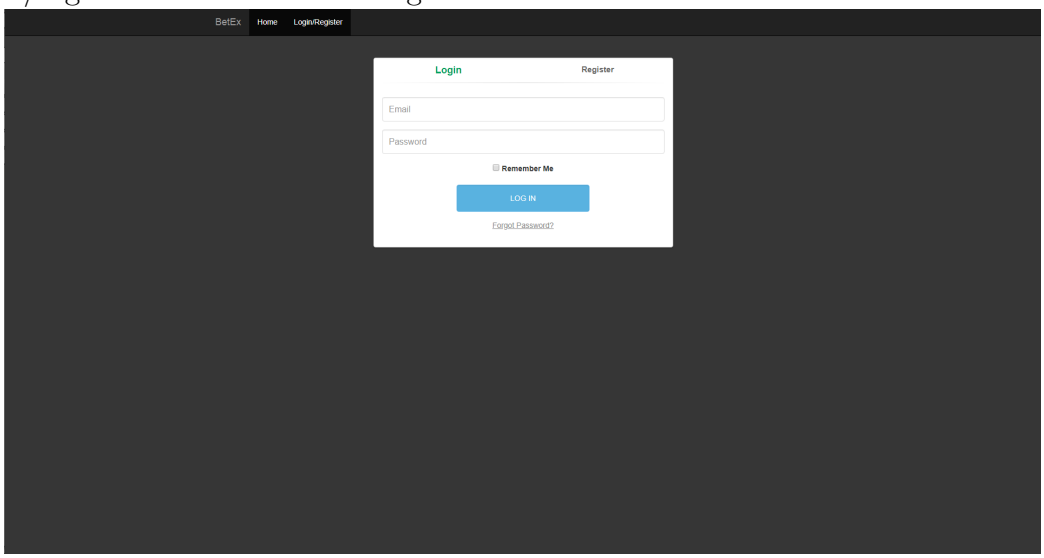
file used to create the database and finally we have the css folder which contains all the css files such as bootstrap.css and style.css, style.css overwrites some of bootstrap.css to make a custom theme.

5.3.2 Pages

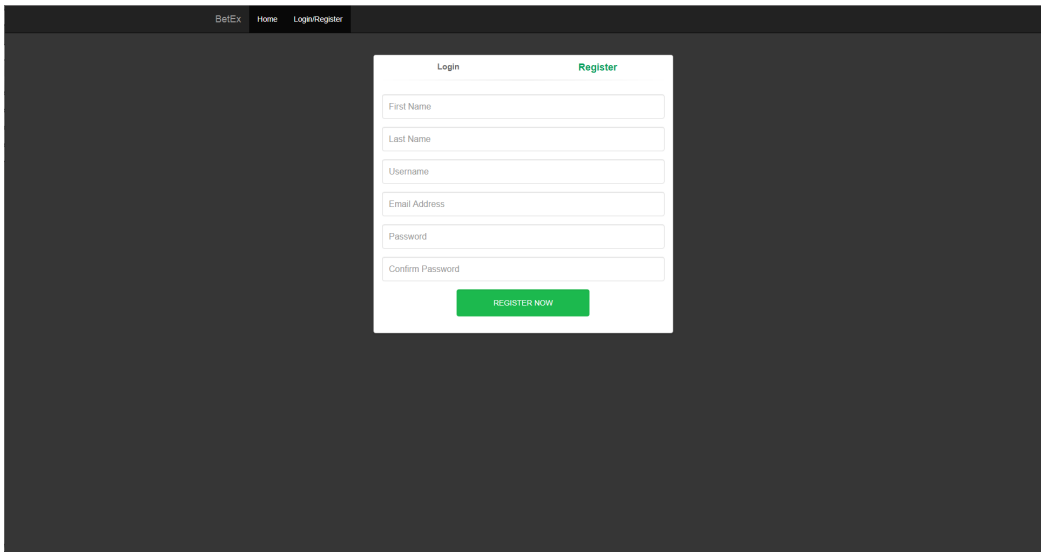
The first picture is of what the user see's when they first visit the site, they are encouraged to sign up in order to unlock the full potential of the site.



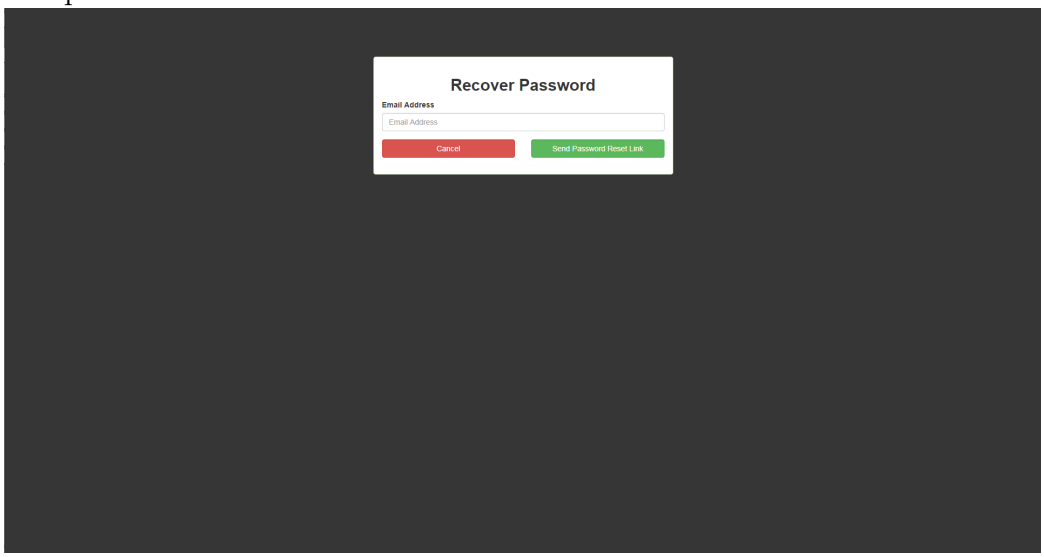
From here the user is then greeted with the log in menu if the press the login/register button on the navigation menu.



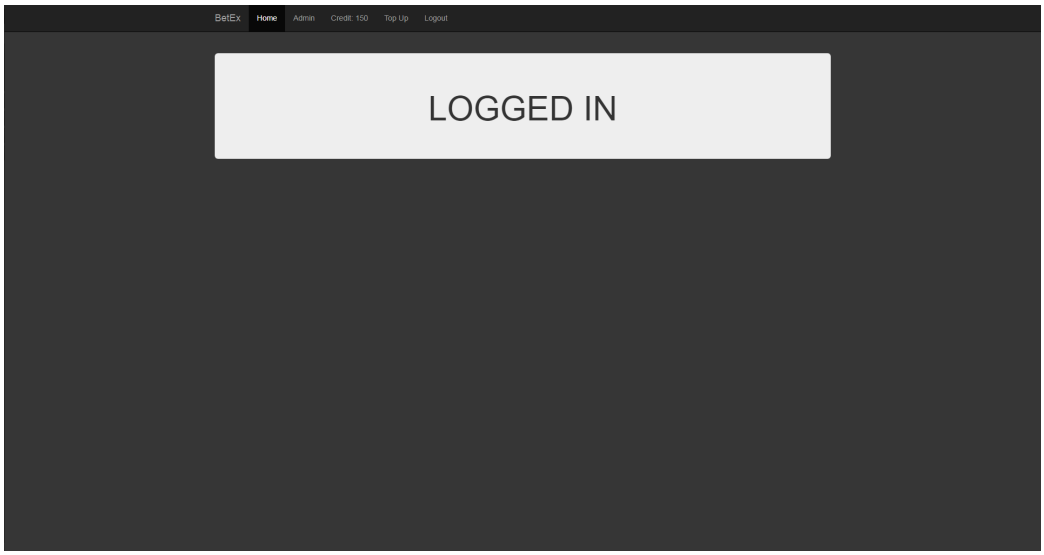
On the other tab in this window is the register form it is similar to the login page.



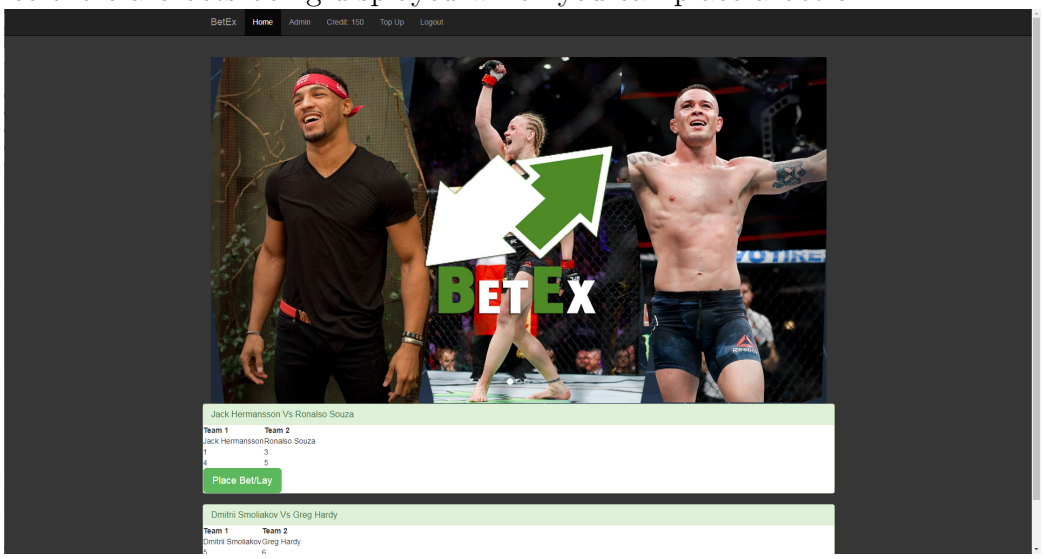
You can also see the forgotten password page from here by clicking the recover password link.



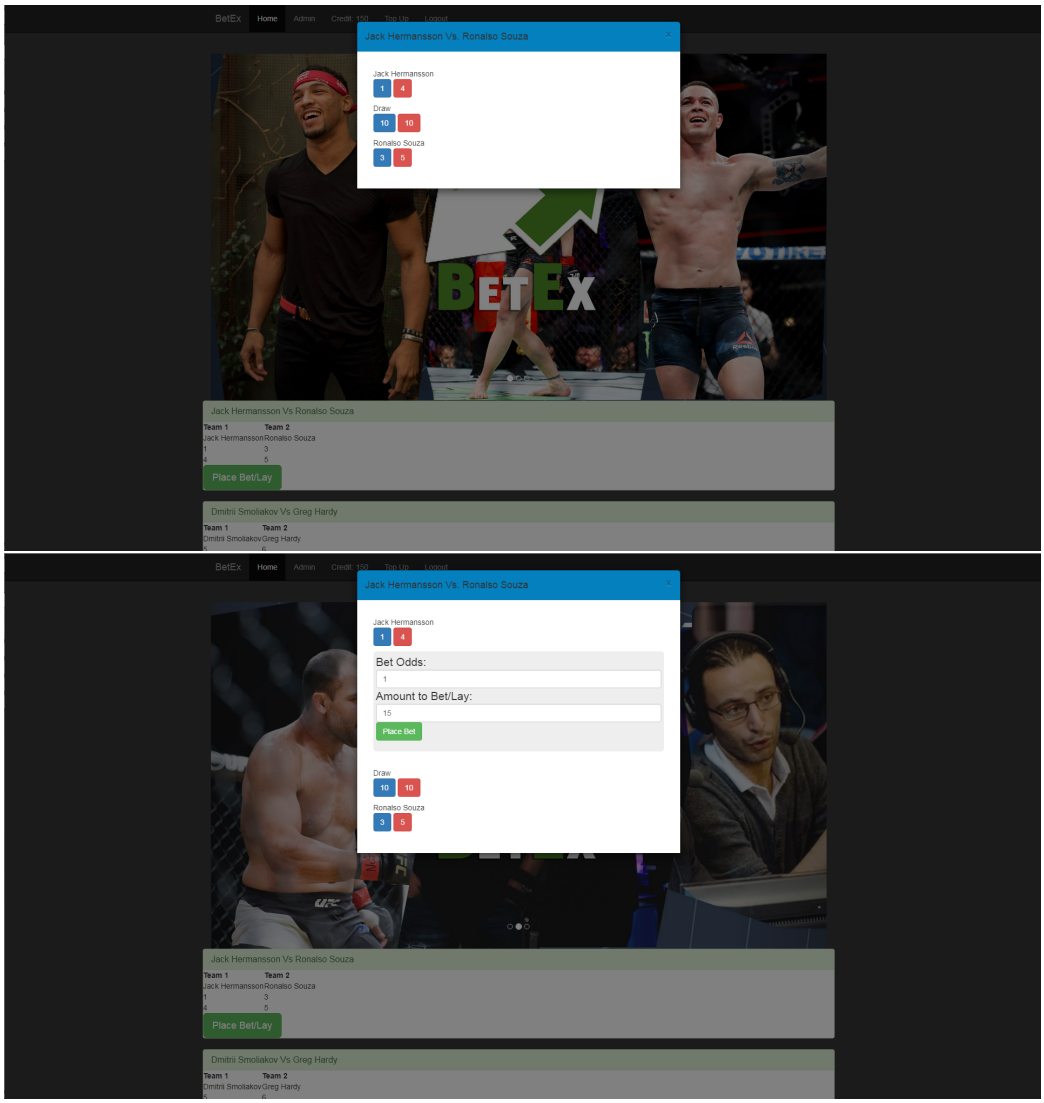
Once the user is logged in it directs the user to their profile page which will contain the history of the bets they have placed.



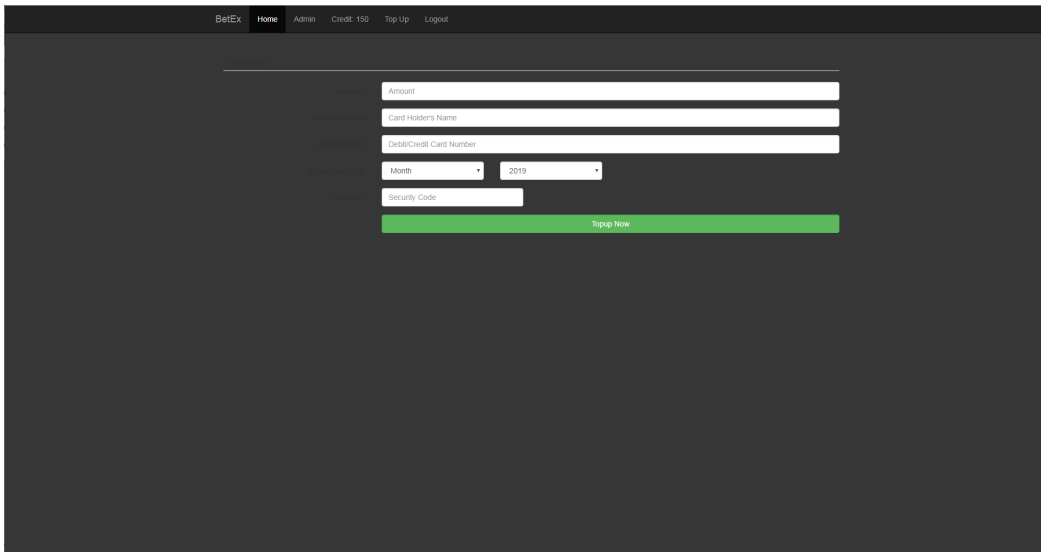
From here you can navigate back to the home page where you will now notice there are bets being displayed which you can place a bet on.



The two images below show the process of placing a bet.



The final page on the site is the top up page which exists to top up the user account when they do not have enough funds.


BetEx
Home
Admin
Credit: 150
Top Up
Logout
Amount
Card Holder's Name
Debit/Credit Card Number
Month
2019
Security Code
Topup Now

Chapter 6

System Evaluation

In this chapter I will evaluate the robustness of the system, the testing and finally the limitations of the application.

6.1 Robustness

Unfortunately due to the oncoming deadline I have not been able to put the application through any real stress testing. However from regular use I have found no real bugs or errors.

6.2 Testing

Through continuous white box testing and regular black box testing I believe the app has reached the expectations I set myself at the very start of the life cycle of this project. In order to complete the white box testing I hosted the site on an Apache server via WAMP and for black box testing via a Google cloud virtual machine.

6.3 Limitations

One of the limitations with this application is that during development I was hosting it locally and could not test out the mail notification system in it's entirety as you had to configure the localhost in a certain way. Another limit of this application is that when it is hosted on the Google cloud virtual machine it is only a static web application and does not represent a fully developed web application.

Chapter 7

Conclusion

In conclusion I can say that this was most definitely the most difficult project I have attempted in my three years in Galway-Mayo Institute of Technology. The scale and complexity of the project has given me greater insight into working on bigger projects in the future. I had no major problems throughout the life cycle of the project, with time and time management being the biggest issue for myself personally.

In the beginning I set out to create a betting exchange platform where users could place bets and lays in a user friendly environment and I can say to a certain extent I have achieved this goal.

What I have accomplished:

- Users can create an account.
- Users can log into their account
- Passwords can be reset
- Logged in users can view their placed bets
- Logged in users can view a list of bets
- Logged in users can place a bet or lay
- Logged in users can top up their account
- List of events is generated via a web scraping python script

7.1 Future Development

There are many avenues one could go in regards to future development within the app. Throughout the development of the application to this point I have

though of many features and possible things to add to the application to improve the experience.

Once such feature might be to implement a versus league where user's can create leagues with their friends and can place bets on opposing sides with a winner take all result.

Another feature is developing the web application into a mobile application. You would then be able to put the app on the app store.

Another feature that I thought of was giving statistics to users such as betting trends displayed in graphic form.

Bibliography

[1] "What is matched betting? the complete guide." https://www.oddsmonkey.com/matched-betting/what-is-matched-betting/.

[2] "Biggest bookies and the distribution of the gambling industry in the uk." https://www.onlinebetting.org.uk/betting-guides/biggest-bookies.html.

[3] "History of php." https://php.net/manual/en/history.php.phpl.