**Lecturer: Dr Leon Smalov**

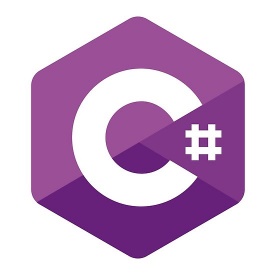
**Module Details:** **M32COM Internet Systems Development**

**Web/Cloud Application Development**

**Produced by:**

**Group H**





**MSC Information Technology**

**Faculty of Engineering, Environment and Computing at Coventry University**



# Introduction

A team of five computing science and IT students have united as a team to undertake a project in establishing a functional web application. Our goal is to design and develop an effective web/cloud application for a remote – control racing boat competition. In this project, we will be expecting to design a front – end interface, retrieving users input to help them to accomplish a task and meet users’ needs. Also, we will be implementing a back-end server which is the SQL database to allow for data to be stored systematically through a logical structure to maintain data integrity, reducing redundancy and allowing data to be stored effectively. We will be storing information such as details of the racing boats, the crew members, competitions relating information, promotional materials, event details, event calendar etc.

Contents

[1. Introduction 2](#_Toc4354729)

[1. Type of a client, programming language 3](#_Toc4354730)

[1.1. Angular 3](#_Toc4354731)

[1.2. C# 3](#_Toc4354732)

[1.3. HTML 3](#_Toc4354733)

[1.4. CSS 3](#_Toc4354734)

[1.5. SQL – Structured Query Language 3](#_Toc4354735)

[1.6. Database: Azure SQL 3](#_Toc4354736)

[1.7. Software 3](#_Toc4354737)

[GitHub 3](#_Toc4354738)

[Visual Studio 4](#_Toc4354739)

[2. Functional prototype 4](#_Toc4354740)

[2.1. Entity Relationship Diagram 4](#_Toc4354741)

[2.2. Use Case 4](#_Toc4354742)

[2.3. Activity Diagram: 5](#_Toc4354743)

[Wireframes - Web application 7](#_Toc4354744)

[3. Provide the complete source code 7](#_Toc4354745)

[4. Documentation and sample set of data for the prototype 7](#_Toc4354746)

[5. Demonstrate the functionality of the prototype application 7](#_Toc4354747)

[6. Conclusion 7](#_Toc4354748)

[7. Reference 7](#_Toc4354749)

# Type of a client, programming language

## Client

## Angular

We used the Angular framework to establish our front-end Web application.

Angular is written in typescript a loosely typed language (mix of static and dynamic) it is compiled down to JavaScript, you can copy JavaScript into a typescript file and it will work, but not the other way round.

It Is a fully-fledged framework, with dependency injection, modules, services, two-way data binding everything needed for a modern enterprise application, mobile and desktop.

As well as the angular framework we user the Angular Material for the look and feel. Angular material is a standard for styling and interacting with an application it has also been optimized so is faster in the browser.

Angular Communicates with the server using Json ( a standard for sending data between client and server), which will allow for uploading data and updating user interface with data received from the server. Angular is open source and is becoming popular amongst web developers.

It is also highly recognized by JavaScript including another framework such as Node.js, React.js, Ember.js etc. They all are programmed in a way of binding data, therefore, is beneficial in displaying dynamically without needing to execute code on the server.

## C#

We have used C#, which is a strongly types language( opposite to javascript but similar to typescript) to object – oriented modem programming language which facilitates and execute code effectively and supports numerous platform, but well recognized by .net frameworks. Using C# for our racing competition web application provides an advantage in design and functionality. C# is deal in implementing good level functionality and features for our racing boat competition web application.

## Java

## HTML

We have used HTML defined as Hyper Mark-up language which is the foundation of maintaining the webpage structure, controlling how content is contained within HTML file structure. HTML is highly effective in communicating with the web browser resulting in displaying an output such as all our related data based on the racing competition through text, images, multimedia on a webpage.

## CSS

We have incorporated CSS stands for cascading Style sheets to help us style and design our front-end interface allowing for a more presentable and interactive web application. CSS allow us to configure the layout format based on front size, border, spacing, height, width, depth, location of HTML structure information on the web pages.

## SQL – Structured Query Language

We had used SQL to help design a relational databases management system allow us to store and manage our data more effectively. Information relating to our remote – control racing boat competition will be stored within our database using SQL scripts and queries based on creation of tables, updating, fetching rows, inserting into and deletion etc.

### Database: Azure SQL

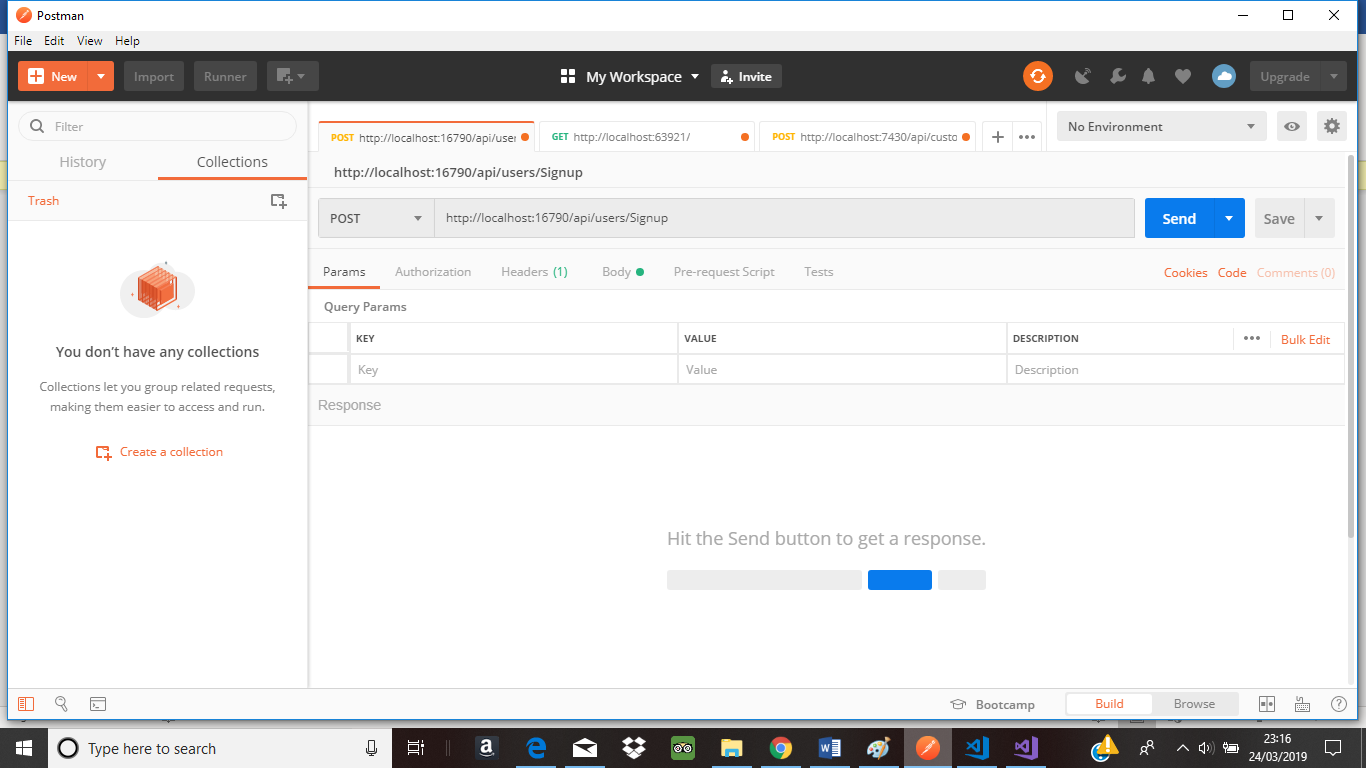
Our racing competition data is stored in Azure SQL database which performs on a high -availability and scalable services. We had previously used Microsoft SQL service management to store our data, unfortunately due to lack of flexibility and due to compatibility and requirements, it was difficult to manage, store data and create a high-level connection to the client server. Azure SQL had resolved our issue which prevented us from installing on our systems, no time-consuming process, no recurring redundancy etc.

## Security – Encryption

Used HTTPS

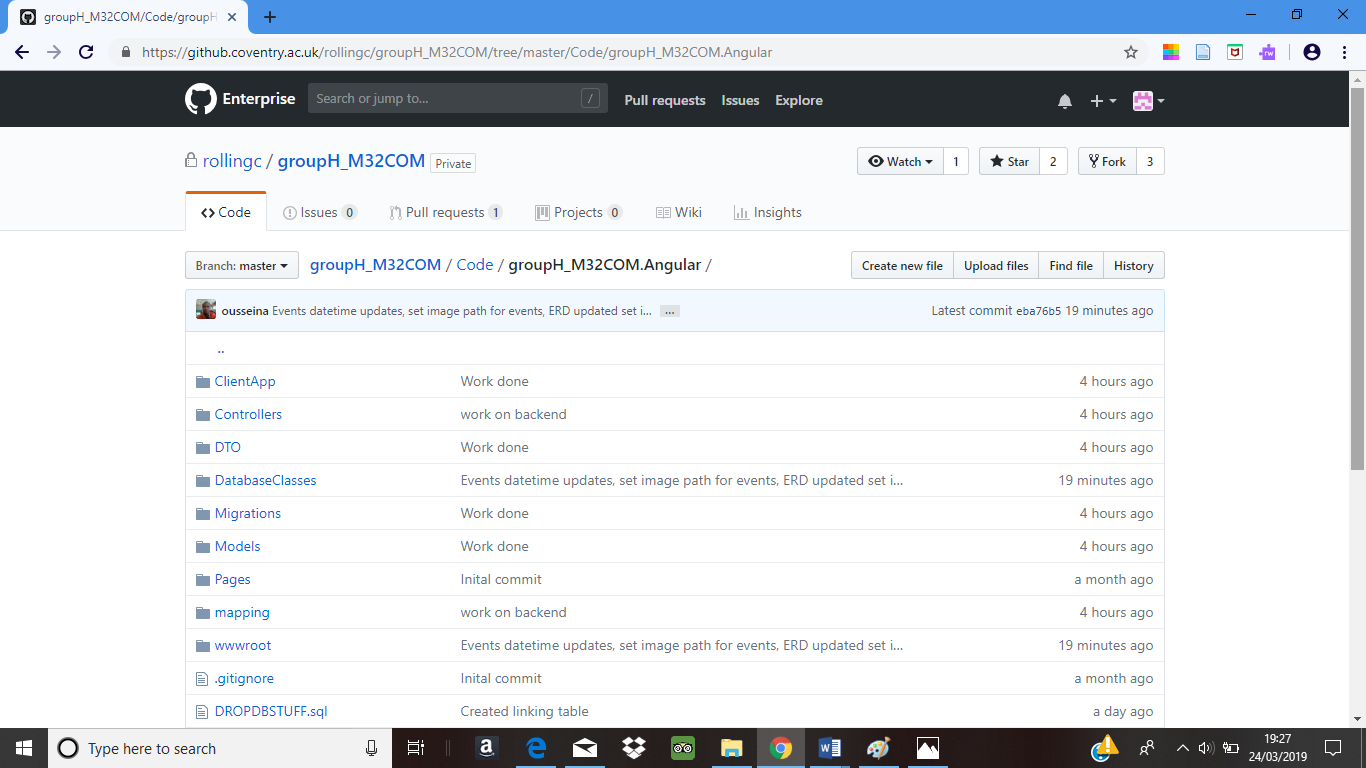
## Software Application

## Postman API

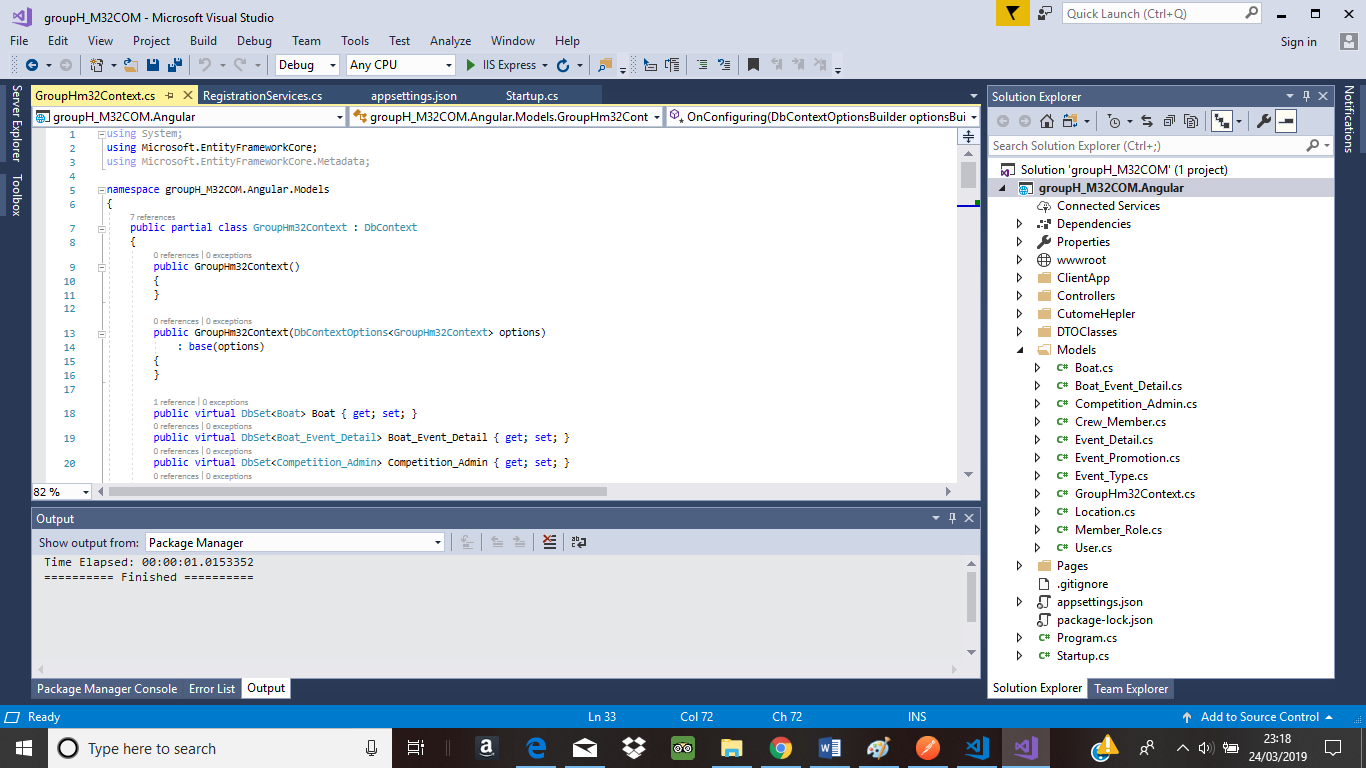
API stands for application programming interface, which is a development tool that is used to test, build, modify APIs. API is handy in checking and testing for communication between the back – end serve and client server. APIs almost allows for any functionality that could by need by any developer is encapsulated in this tool. APIs are capable of processing various HTTP requests such as GET, POST, PUT, PATCH. We had used an API to test and build our authorization for our registration and login form.

## GitHub

GitHub provides flexibility allowing users to edit and interact with code from a Git repository hosting service. Git supports many functionalities and features to support user’s system application and needs. It contains its own command line tool, a web – based graphical interface. Allows users to set privileges meaning users can control which users can gain access to the code and add multiple of users to collaborate with, without needing to download the entire application on their system.

GitHub is also an effective tool in managing file structure, has benefitted our group in editing, locating the root folder, updating existing code, diagrams, structure, documents etc.

## Visual Studio

We have used Visual studio program to build and deploy our racing competition web application, as you are aware we had implemented numerous meaningful programming language such as Angular, C#, SQL, MYSQL, HTML etc. Therefore, using visual studio was beneficial as it supported our web design and functionality in implementing, compiling, executing the code successfully. Visual contains error handling and debugging tool allowing us to identify errors within our code along with guidance on how to problem solve that error.

## ASP.NET Framework

## Data types Used

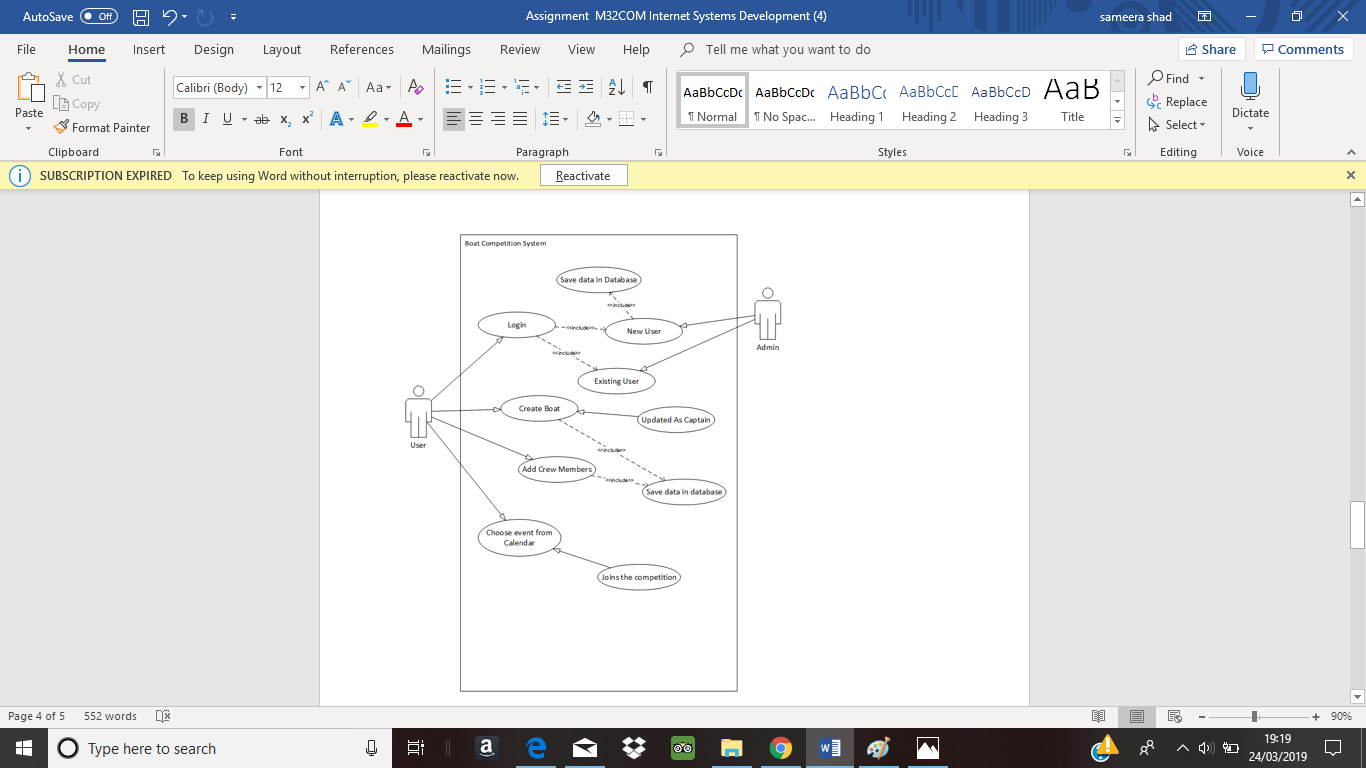
|  |  |  |  |
| --- | --- | --- | --- |
| Data Types | Description | Example of Data | Typical Use |
| Text/ Varchar | Characters |  |  |
| Integer | Whole Number |  |  |
| String |  |  |  |
| Date | Time |  |  |
| Boolean | True or False |  |  |
| Image | A graphic file |  |  |

# Functional prototype

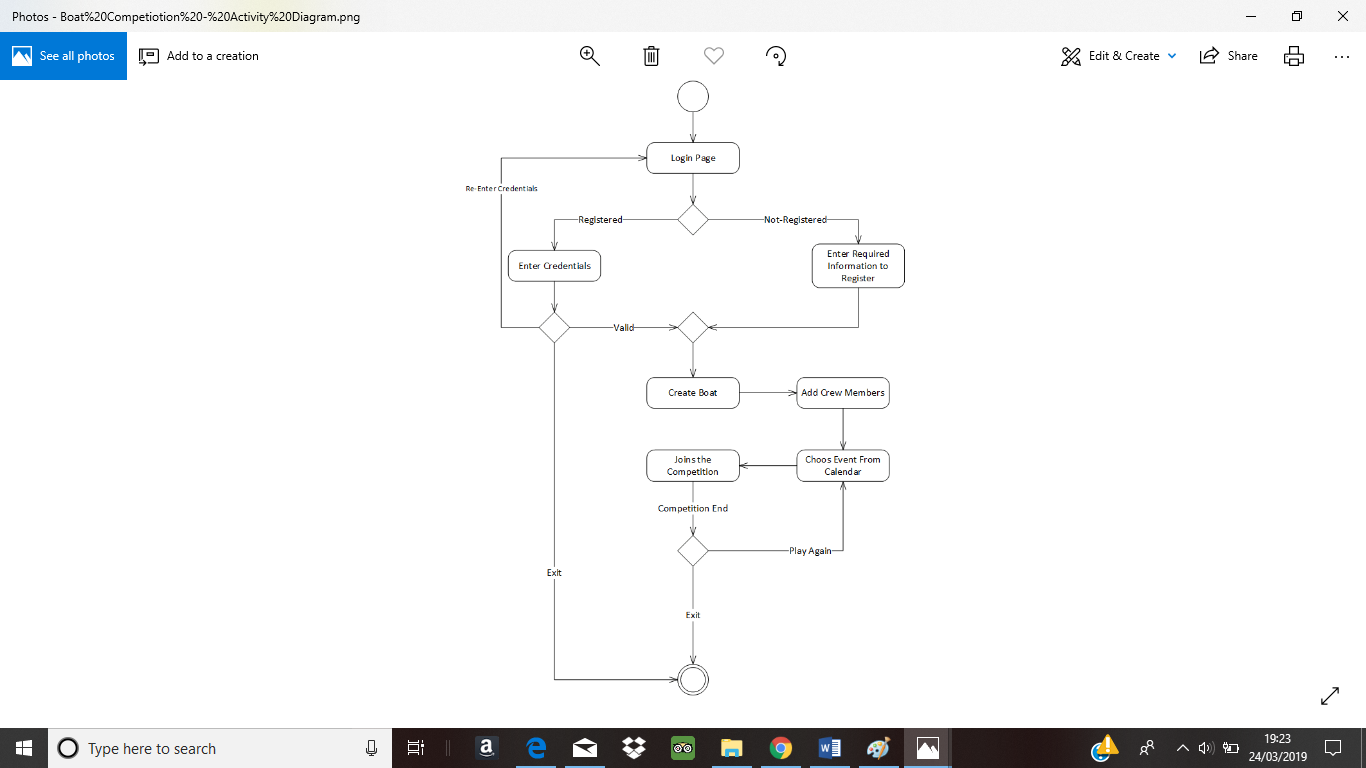
## Sitemap

## Use Case

* User Go to the login page.
* If the user is not registered, he must register himself to the Portal.
* After logging-in, the user Creates the boat which makes him the “Captain” of the boat.
* Then the user Adds his Crew members.
* Boat data and crew members data is saved in database.
* Then the User chooses any event from calendar and joins the competition.



## Activity Diagram:

* User Go to the login page.
* If the user is not registered, he must register himself to the Portal.
* While logging-in, if the credentials are not valid, user is redirected to the login page again.
* After logging-in, the user Creates the boat.
* Then the user Adds his Crew members.
* Then the User chooses any event from calendar and joins the competition
* After the competition is ended, the user has option to choose event and play again or may exit from the system.

## Wireframes - Web application

## Front – end – Client server Design

### Index PAGE – Home Design

### 

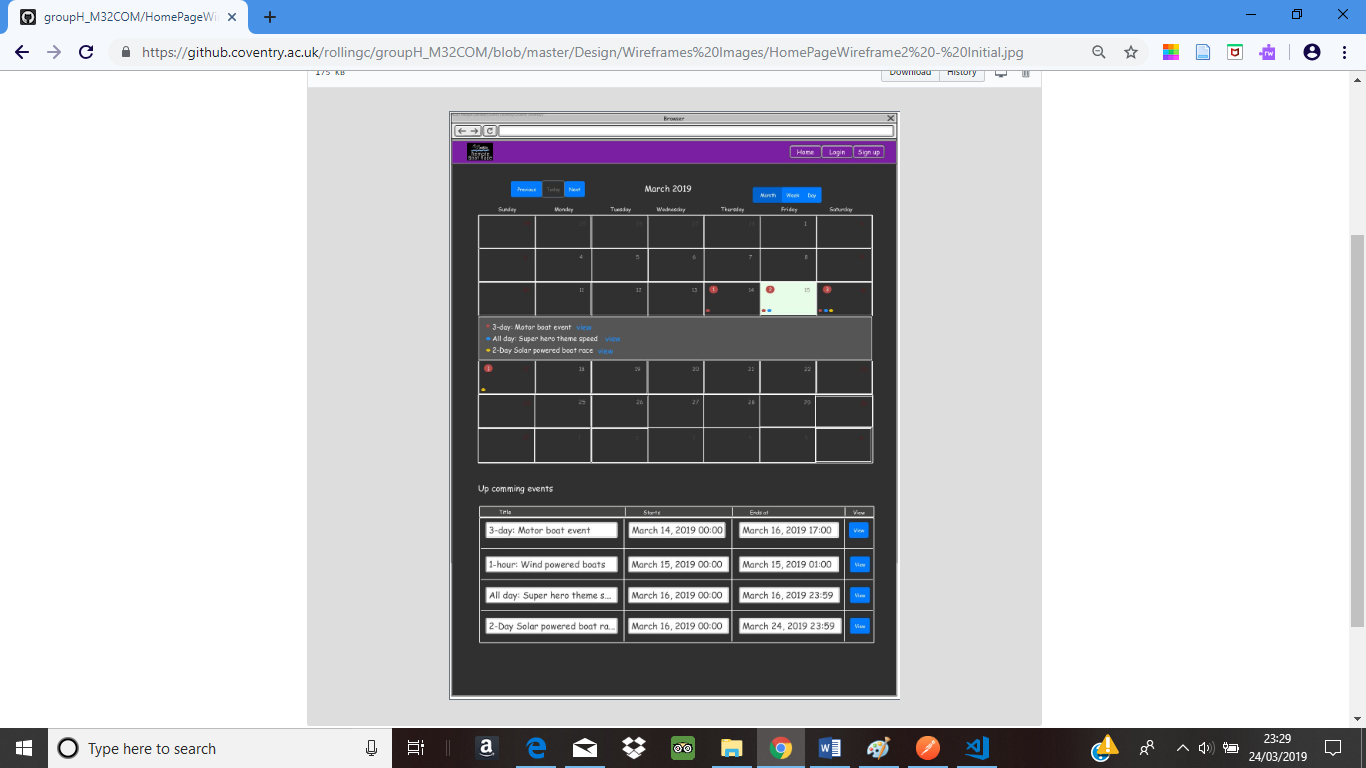
Navigation Bar

**Tabs**

Calendar

**Buttons**

Headings

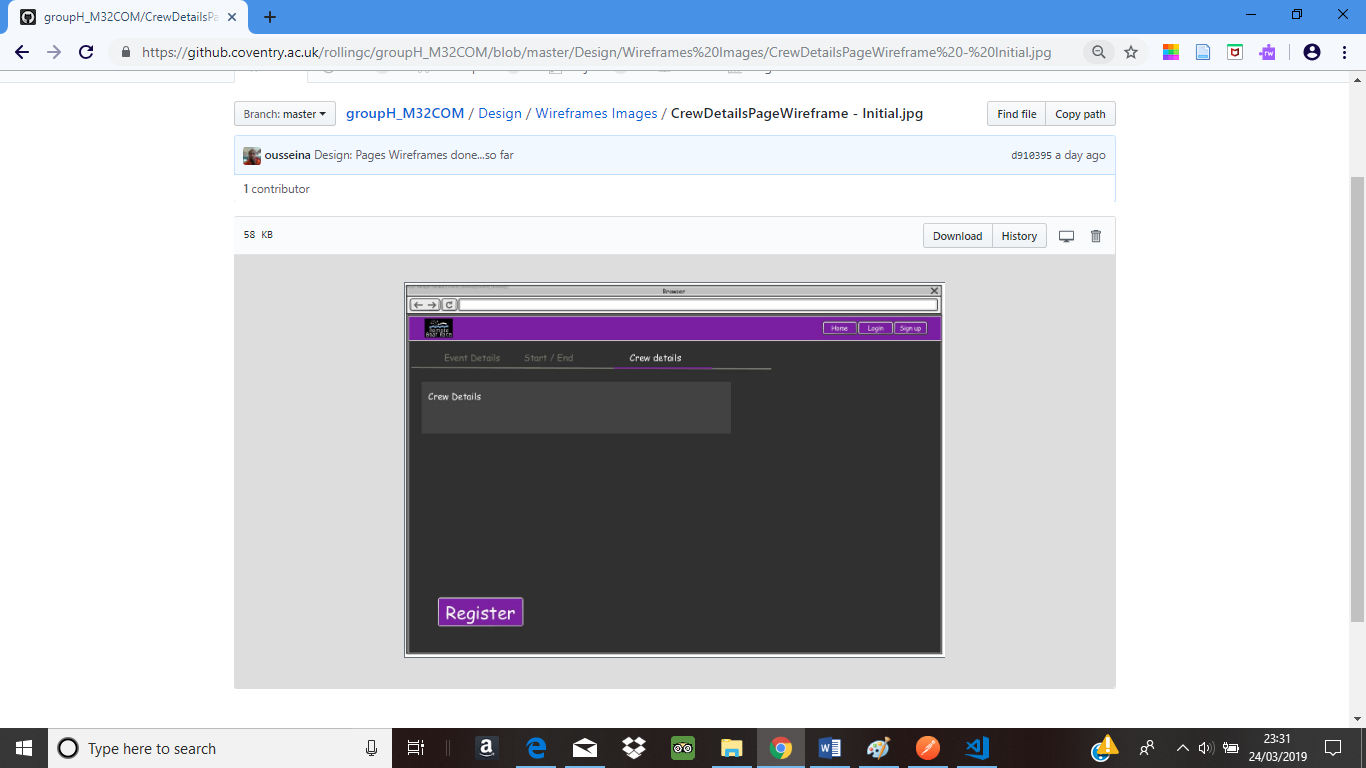


Logo

**Text**

**Text Field - Input**

### Crew Details Design

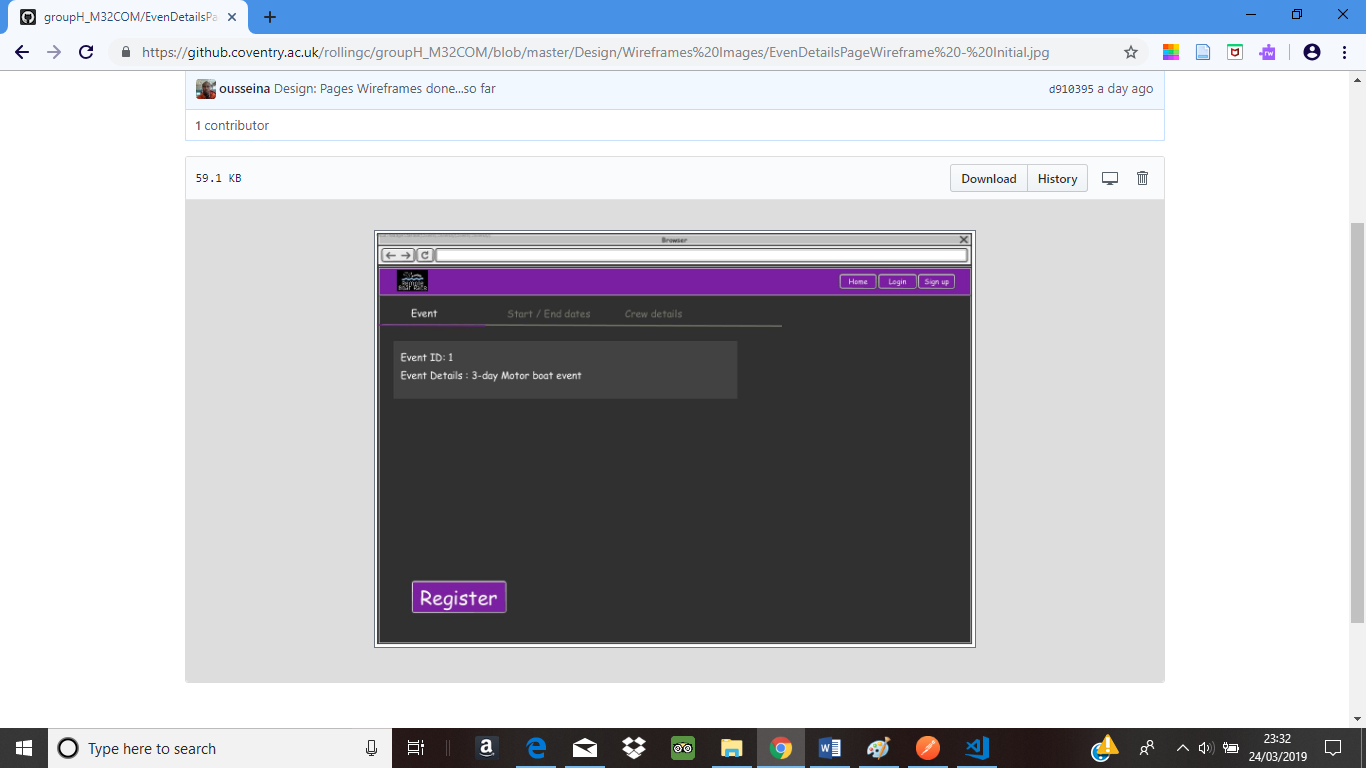


**Buttons**

**Contentnt**

**Tabsnt**

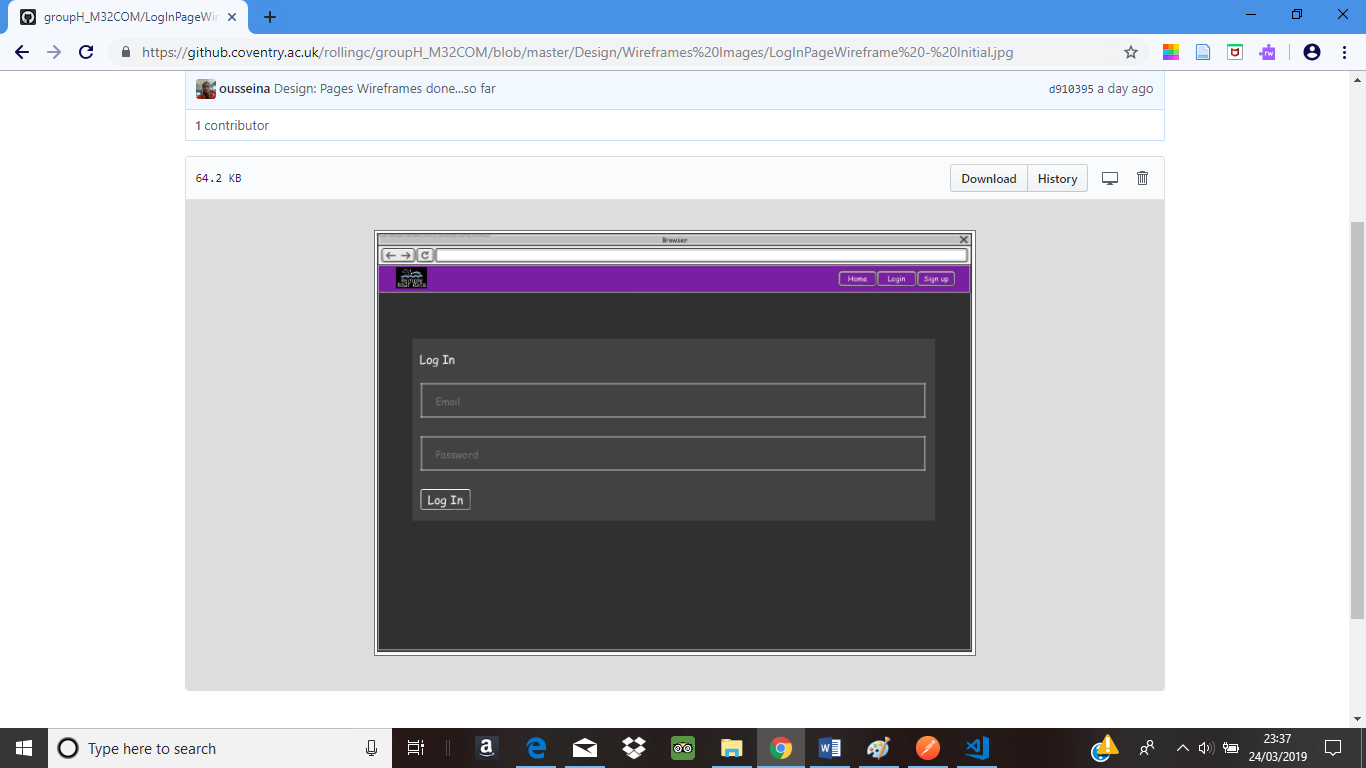
### Event Page Design



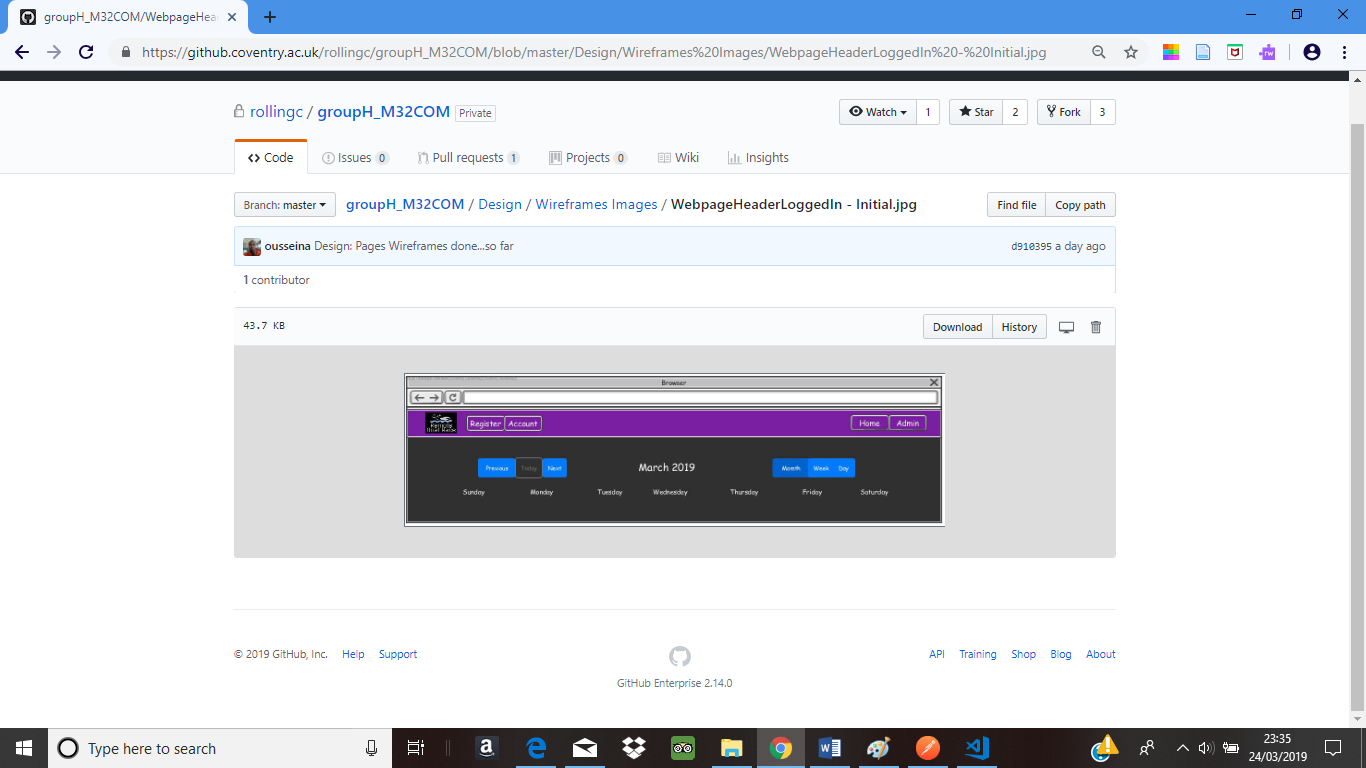
### Sign up Design

# 

### Login in Design



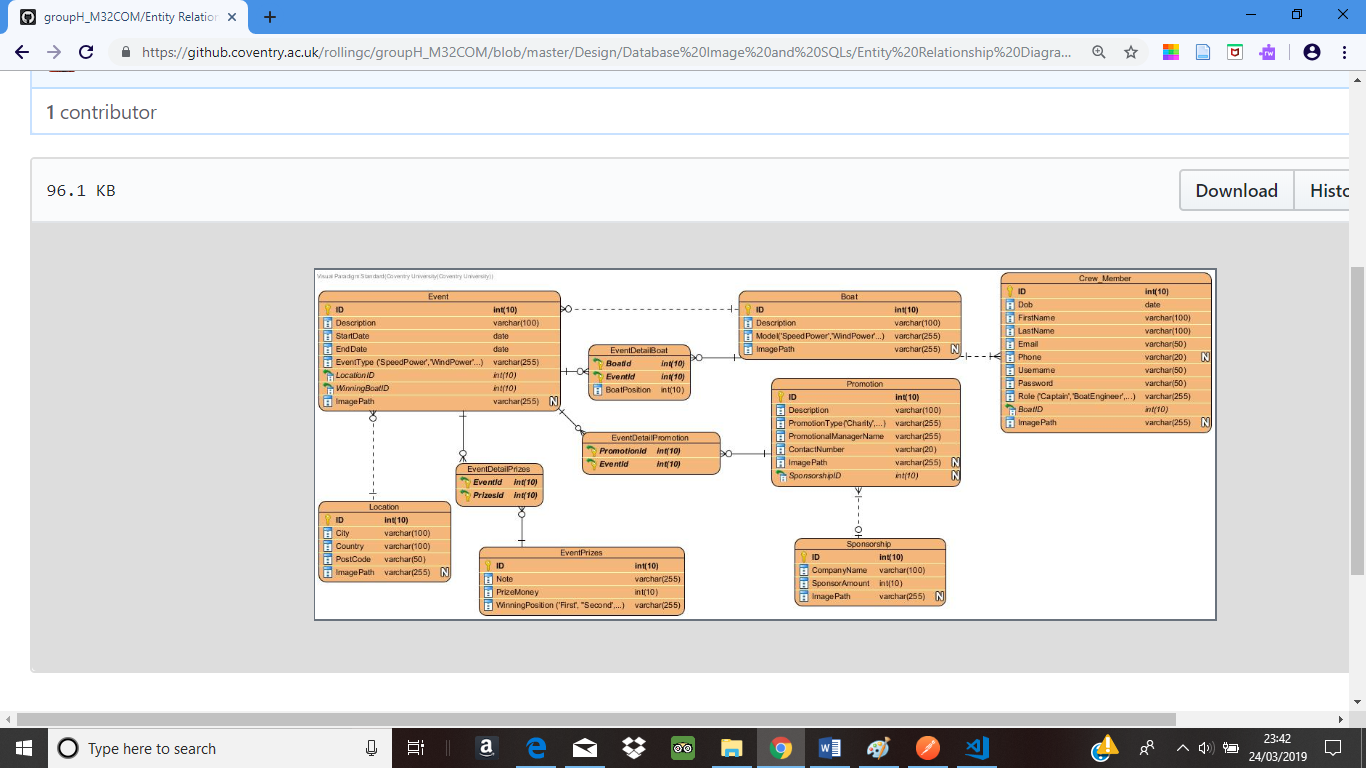
### Header Design



## ERD – Entity Relationship Diagram

### Back – end Server – Database Design

* Crew Members can have one to many Boats
* Boat can only be driven by one crew member only
* Each Event requires a boat for each event only
* 0 to many events can be taken place in one location
* Each event can contain up to one to many Prizes
* Each event can have up to 0 to many promotions
* Promotion can receive up to zero to one Sponsorships
* Sponsorship can receive up to one to many promotions



## UML Diagram

# Provide the complete source code

1. Documentation and sample set of data for the prototype

# Demonstrate the functionality of the prototype application

# Conclusion

# Reference