

**Summer Project On
Football Blog Application
By
Kshitij Koyande (2021510029)**

Under the guidance of
Internal Supervisor

Prof. Pallavi Thakur



Department of Master Of Computer Application
Sardar Patel Institute of Technology
Autonomous Institute Affiliated to Mumbai University
2022-23

CERTIFICATE OF APPROVAL

This is to certify that the following students

Kshitij Koyande (2021510029)

Have satisfactorily carried out work on the project
entitled

“Football Blog Application”

Towards the fulfilment of project, as laid down
by
Sardar Patel Institute of Technology
during year
2022-23.

Project Guide:
Prof. Pallavi Thakur

PROJECT APPROVAL CERTIFICATE

This is to certify that the following students

Kshitij Koyande (2021510029)

Have successfully completed the Project report on

“Football Blog Application”,

which is found to be satisfactory and is approved

at

SARDAR PATEL INSTITUTE OF TECHNOLOGY,
ANDHERI (W), MUMBAI

INTERNAL EXAMINER

EXTERNAL EXAMINER

HEAD OF DEPARTMENT

PRINCIPAL

Contents

Abstract	i
Objectives	i
List Of Figures	ii
List Of Tables	ii
1 Introduction	1
1.1 Problem Definition	1
1.2 Objectives and Scope	1
1.2.1 Objectives	1
1.2.2 Scope	1
1.3 Existing System	2
1.4 Proposed System	2
1.5 System Requirements	3
2 Software Requirement Specification (SRS) and Design	4
2.1 Purpose	4
2.2 Definition	4
2.3 Overall Description	4
2.3.1 Product Functions	4
2.3.2 User Characteristics	5
3 Project Analysis and Design	6
3.1 Methodologies Adapted	6
3.2 Modules	7
3.2.1 Activity diagram	7
3.2.2 Communication Design	8
3.2.3 Work Breakdown Structure	8
3.2.4 PERT Chart	9
3.2.5 Gantt Chart	9
3.2.6 Use-Case	10
4 Project Implementation and Testing	13
4.1 Login and Register	14
4.2 Home - Landing View	15
4.3 Posts	16
4.4 Write Posts	17
4.5 Code 1	18
4.6 Code 2	18
4.7 Code 3	19
5 Test Cases	20

6	Limitations	21
7	Future Enhancements	21
8	User Manual	22
9	Bibliography	23
9.1	Web References	23

Abstract

Football blog is a blogging website which provides articles on football for users to read. It is a website which has all the CRUD functionality and is created using React for front end and Node and Express for backend. It uses MongoDB as a database.

The User can read articles on football. The user can also write their own blogs. He can update and delete his own blogs. He can also like aswell as comment on other users' blogs.

Objectives

The Web Application "Football blog" is used

- To provide a user friendly platform for users to read about football.
- To allow users to write their own blogs

List of Figures

3.1.1Diagrammatic Representation of Waterfall Model	6
3.2.1Activity Diagram	7
3.2.2Communication Diagram	8
3.2.3Work Breakdown Structure	8
3.2.4PERT Chart	9
3.2.5Gantt Chart	9
3.2.6Use-Case Diagram	10
4.1.1 Login and Register	14
4.2.1 Home View	15
4.3.1Profile	16
4.4.1Write Posts	17

List of Tables

1.5.1 Hardware Requirements on Server Side	3
1.5.2 Hardware Requirements on Client Side	3
1.5.3 Software Requirements on Server Side	3
1.5.3 Software Requirements on Client Side	3
4.2.1 Use Case Table - Register	11
4.2.2 Use Case Table - Login	11
4.2.3 Use Case Table - Update Profile	11
4.2.4 Use Case Table - Update Posts	12
4.2.5 Use Case Table - View Posts	12
6.1 Test Case - Login and Register	20
6.2 Test Case - Others	20

1 Introduction

1.1 Problem Definition

To create an application through which users can get indepth coverage about the sport.

1.2 Objectives and Scope

1.2.1 Objectives

The Web application "Football Blog" is

- To provide in depth insights about football to the user.
- To create a community of people interested in the sport
- To provide users to see live match scores.

1.2.2 Scope

The user can provide his/her details in the profile and view for various blog posts on the app.

In the application user must enter his/her personal details in the profile section.

Our System is being made for creating a community of like minded people who can share their opinions about the post.

1.3 Existing System

Currently there are no football specific websites which are focused on the users opinion. All websites that currently exists are news applications which have their own authors giving their opinion. There is no such blogging platform specific to the sport

1.4 Proposed System

The User will be able to register and login to the application. After logging into the system the user will be able to read the blog posts. He can also update or delete his own created posts. He can also like and comment on other people's posts. The posts are classified according to various sections and competitions. The user can also search for posts using search bar. If a user has not logged in then he can only read other person's posts. The user can upload and update his profile name and profile picture. He can also see live scores of all the matches that are ongoing right now.

1.5 System Requirements

- Hardware Requirements on Server Side

Table 1.5.1: Hardware Requirements on Server Side

Processor	Dual Core Processor or Above
RAM	Minimum 4 GB RAM
Storage	Minimum 10 GB Hard Disk Space

- Hardware Requirements on Client Side

Table 1.5.2: Hardware Requirements on Client Side

Laptop or PC or Mobile	
Processor	Dual Core Processor or Above
RAM	Minimum 2 GB RAM
Storage	Minimum 250 MB Storage Space

- Software Requirements on Server Side

Table 1.5.3: Software Requirements on Server Side

Operating System	OS Independent
Database	MongoDB

- Software Requirements on Client Side

Table 1.5.3: Software Requirements on Client Side

Operating System	PC/Android/IOS Smartphone
Server	Not Required

2 Software Requirement Specification (SRS) and Design

2.1 Purpose

The purpose of this project is to create a blog based application which is specific to football. The user can write their own blogs. They can also update and delete their own blogs. It provides a platform for all the football fans to come together and write give their own opinions through writing blogs.

2.2 Definition

To build a blog application about football

2.3 Overall Description

2.3.1 Product Functions

The product function includes:

1. Authentication: Users are required to Sign-up and Log-in .
2. Profile: This will contain information regarding the user.
3. Write: This allows the users to write their own posts.
4. Update: Users can update only the posts they have written.
5. Delete:Users can delete only the posts they have written .
6. Like:Users can like other people's posts.
7. Comment:Users can comment on other people's posts.
8. UpdateProfile:Users can update their own profile picture and profile information.

2.3.2 User Characteristics

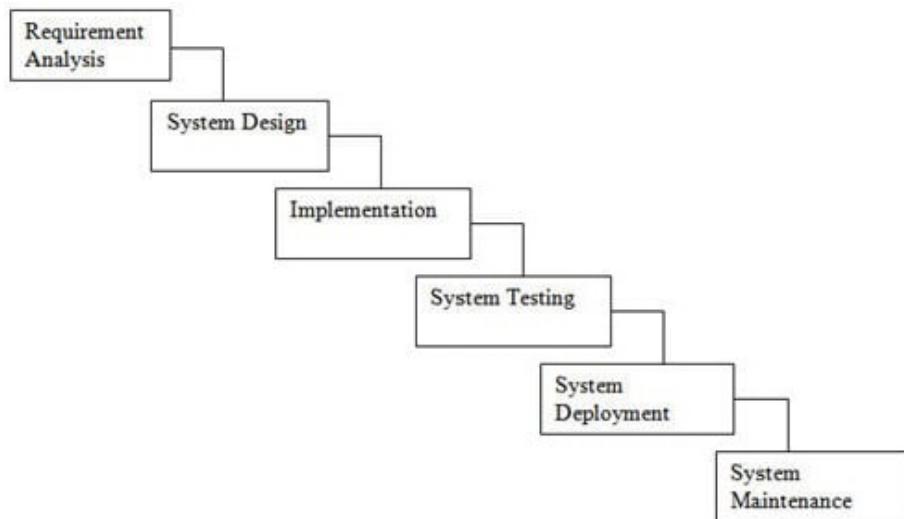
- Application User: Once the user has logged in he can write, update and delete their own posts .
- The user can also read , like and comment on other people's posts.

3 Project Analysis and Design

3.1 Methodologies Adapted

In Waterfall model, very less customer interaction is involved during the development of the product. Once the product is ready then only it can be demonstrated to the end users.

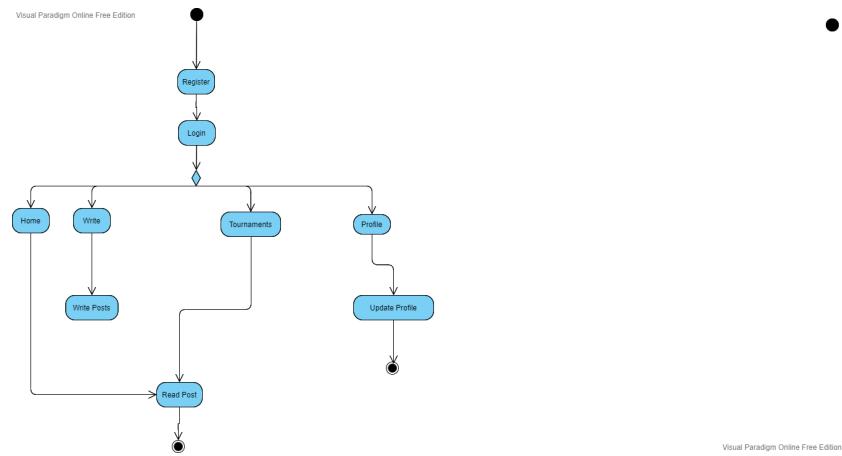
Once the product is developed and if any failure occurs then the cost of such issues is very high, because we need to update everything from document till the logic.



3.1.1: Diagrammatic Representation of Waterfall Model

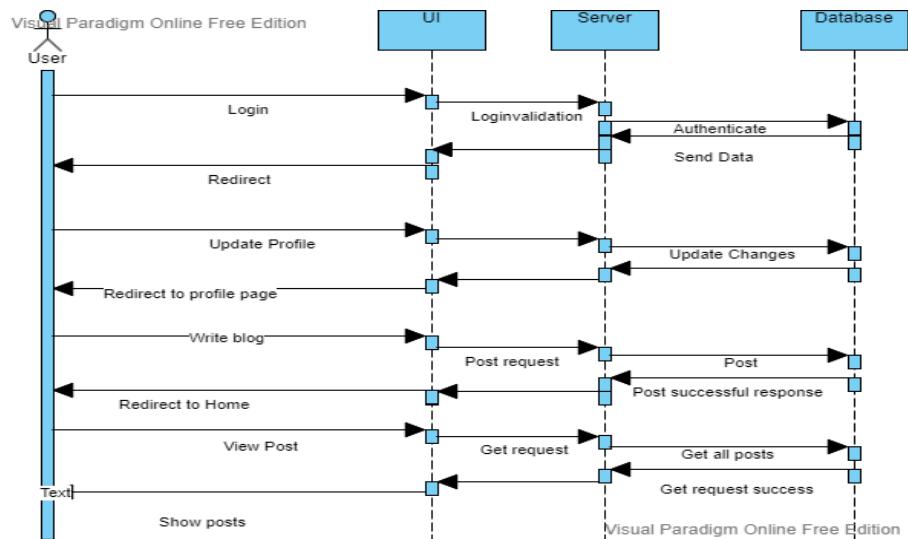
3.2 Modules

3.2.1 Activity diagram



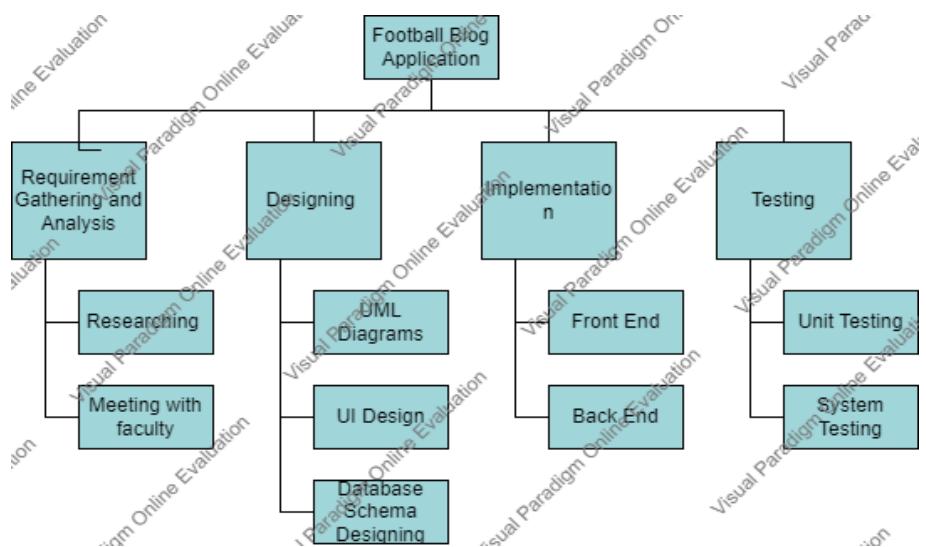
3.2.1: Activity Diagram

3.2.2 Communication Design



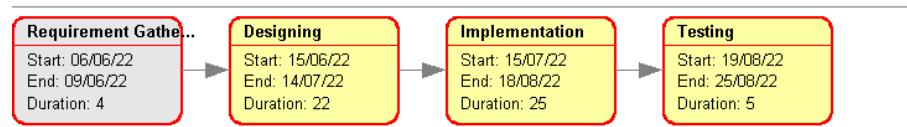
3.2.2: Communication Diagram

3.2.3 Work Breakdown Structure



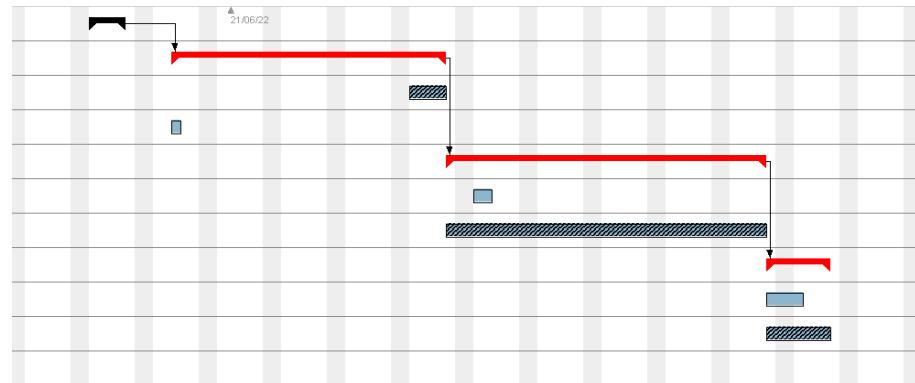
3.2.3: Work Breakdown Structure

3.2.4 PERT Chart



3.2.4: PERT Chart

3.2.5 Gantt Chart

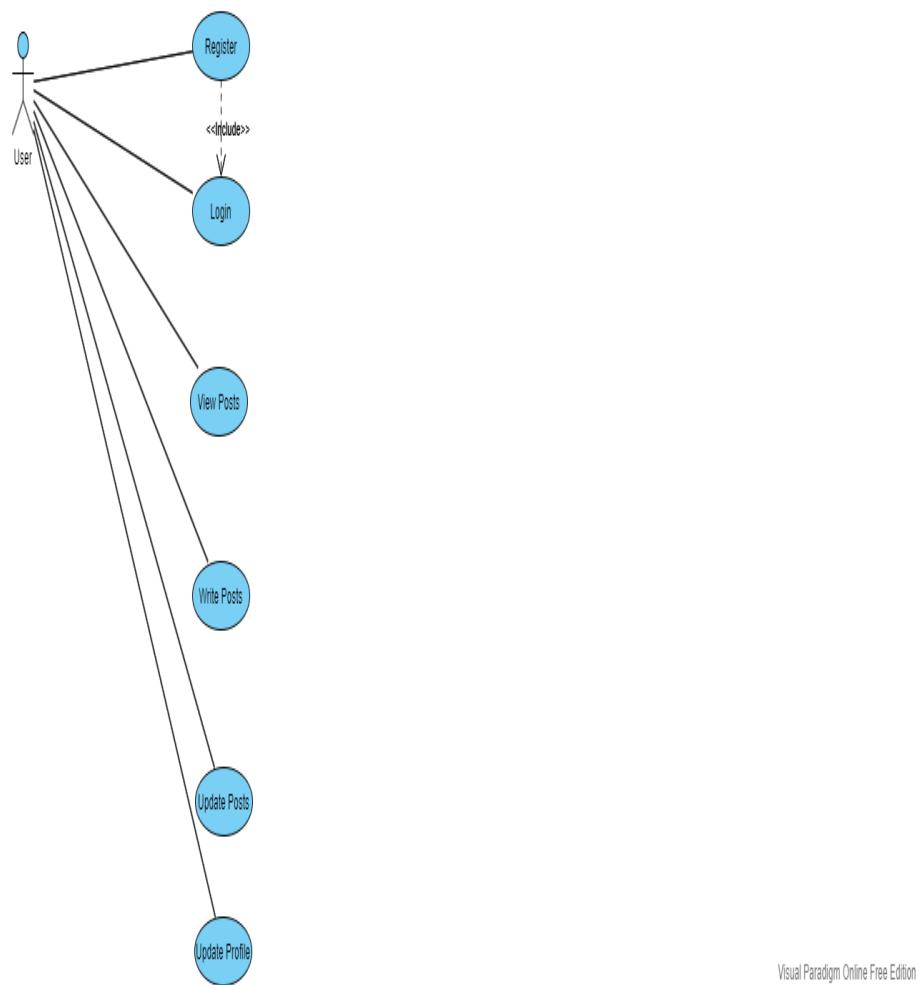


3.2.5: Gantt Chart

3.2.6 Use-Case

Visual Paradigm Online Free Edition

<br class="Apple-interchange-newline">



Visual Paradigm Online Free Edition

3.2.6: Use-Case Diagram

Use Cases:

1. Register
2. Login
3. View Posts
4. Update Posts
5. Write Posts
6. Update Profile

Table 4.2.1: Use Case Table - Register

Use Case ID	1
Use Case Name	Register
Actor	User
Pre-Condition	They must register themselves first
Post-Condition	User can login
Flow of events	Login,Register

Table 4.2.2: Use Case Table - Login

Use Case ID	2
Use Case Name	Login
Actor	User
Pre-Condition	They must register themselves first
Post-Condition	User can view his own profile,view and write posts
Flow of events	Login,View or Write Posts

Table 4.2.3: Use Case Table - Update Profile

Use Case ID	3
Use Case Name	Update Profile
Actor	User
Pre-Condition	Login
Post-Condition	User can view or edit the profile

Table 4.2.4: Use Case Table - Update Posts

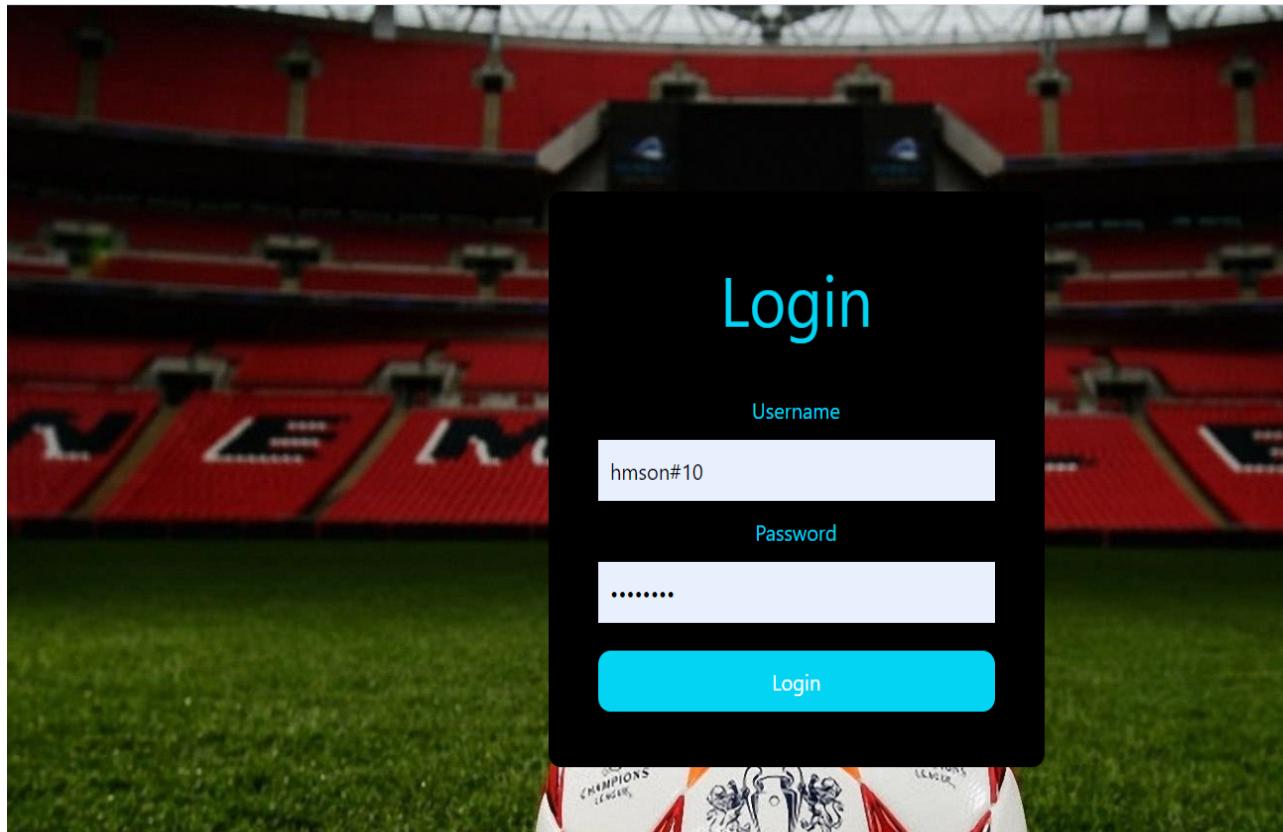
Use Case ID	4
Use Case Name	Update Posts
Actor	User
Pre-Condition	Login and upload the post
Post-Condition	User can view and edit the post

Table 4.2.5: Use Case Table - View Posts

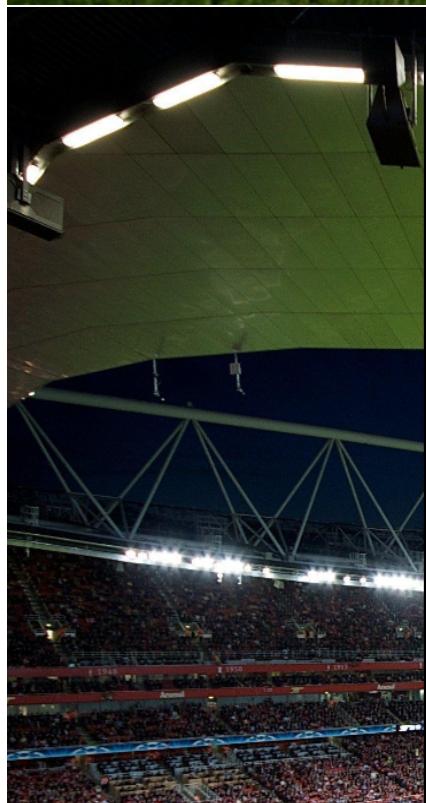
Use Case ID	5
Use Case Name	View Posts
Actor	User
Pre-Condition	Login
Post-Condition	Can view posts

4 Project Implementation and Testing

4.1 Login and Register



The background of the entire image is a photograph of a football stadium at night, showing the illuminated pitch and the stands filled with spectators.



The left side of the registration form has a vertical background image of a football stadium at night, showing the illuminated pitch and the stands filled with spectators.

Login

Username

Password

Login



Register

Email ID

Enter your Email ID

Not a valid email id

Username

Password

Page 14

Confirm Password

Reenter password

4.2 Home - Landing View

Football Blog

Get Latest Insights About Everything Football

Home Write Premier Champions League Login

Q



Cristiano Ronaldo transfer:
Bayern Munich favourite to
sign Man United star

kloppanfield

Sun Jul 03 2022



Bayern to sell winger

kloppanfield

Sun Jul 03 2022



Will Haaland's arrival change
the way Manchester City play

kloppanfield

Page 15
Mon Jul 04 2022



gnabry to leave

kloppanfield

Mon Jul 04 2022

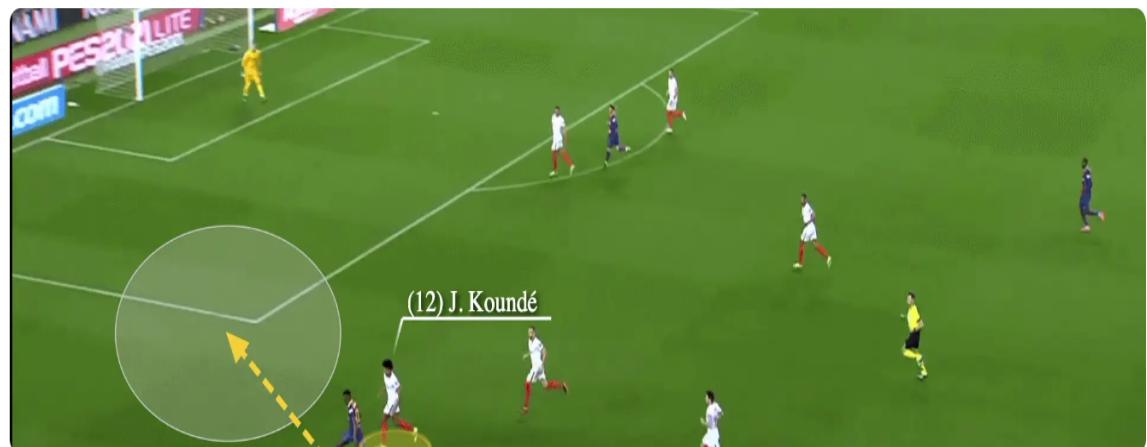
4.3 Posts

The screenshot shows a blog post interface. At the top, there are social media sharing icons for Instagram and Twitter, followed by navigation links: Home, Write, Premier, Champions League, and Login. Below this is a large, blurred image of a soccer ball. The main title of the post is "Robert Lewandowski to Barcelona: Bayern will miss the goals but not the drama". Underneath the title, the author is listed as "klopp" and the date is "Sun Jul 24 2022". There are 5 likes indicated. The post content begins with a paragraph about the emotional atmosphere at the Allianz Arena in May 2019 during Bayern Munich's last home game before the transfer of Arjen Robben and Franck Ribery.

In May 2019, the Allianz Arena crowd was awash with emotion as joy and sadness pulled at opposite ends of the heartstrings. Club legends Arjen Robben and Franck Ribery starred in their last home game for Bayern Munich, a 5-1 win over Eintracht Frankfurt that secured their seventh championship in a row. Club president Uli Hoeness was one of many supporters crying in the stands amid the general outpouring of love, gratitude and regret that greeted the iconic duo's departure. Even Robert Lewandowski, a man who can rarely be accused of excess sentimentality,

4.3.1: Profile

4.4 Write Posts



My first post

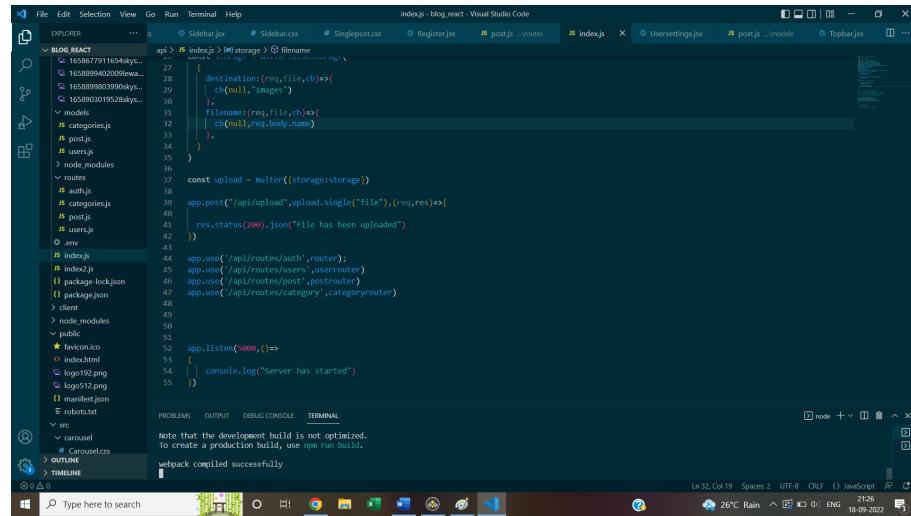
This is my first post

4.4.1: Write Posts

Football Blog Application

Kshitij Koyande (2021510029)

4.5 Code 1



A screenshot of Visual Studio Code showing the code for handling file uploads. The code is located in the `index.js` file under the `routes` directory. It uses the `multer` middleware to handle file uploads and the `multerStorage` function to store files in the `images` folder. The code then handles the uploaded file and returns a success response.

```
const upload = multer({storage:storage})

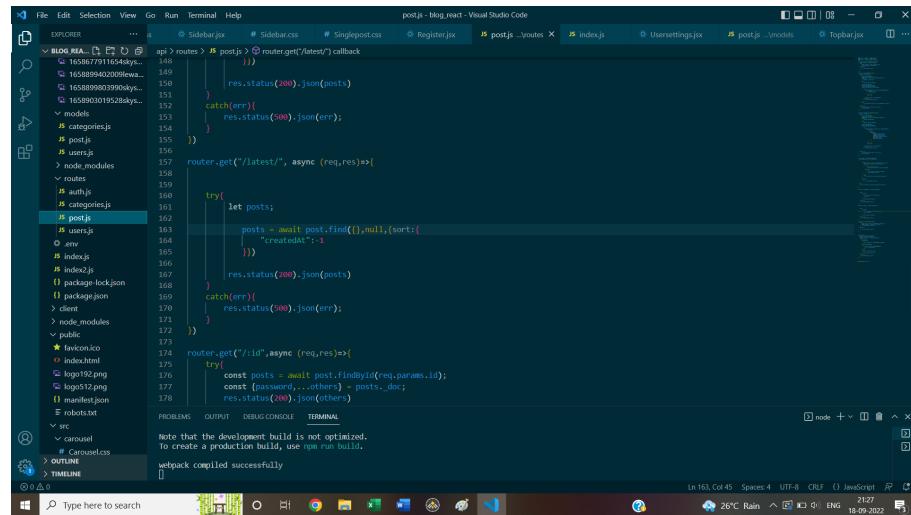
app.post('/api/upload',upload.single('file'),(req,res)=>
  res.status(200).json("file has been uploaded")
)

app.use('/api/routes/auth',router);
app.use('/api/routes/users',userRouter)
app.use('/api/routes/post',postRouter)
app.use('/api/routes/category',categoryRouter)

client
node_modules
public
  favicon.ico
  index.html
  logo192.png
  logo512.png
  manifest.json
  robots.txt
src
  carousel
    Carousels
  OUTLINE
  TIMELINE

Type here to search 2126 26°C Rain 18-09-2022
```

4.6 Code 2

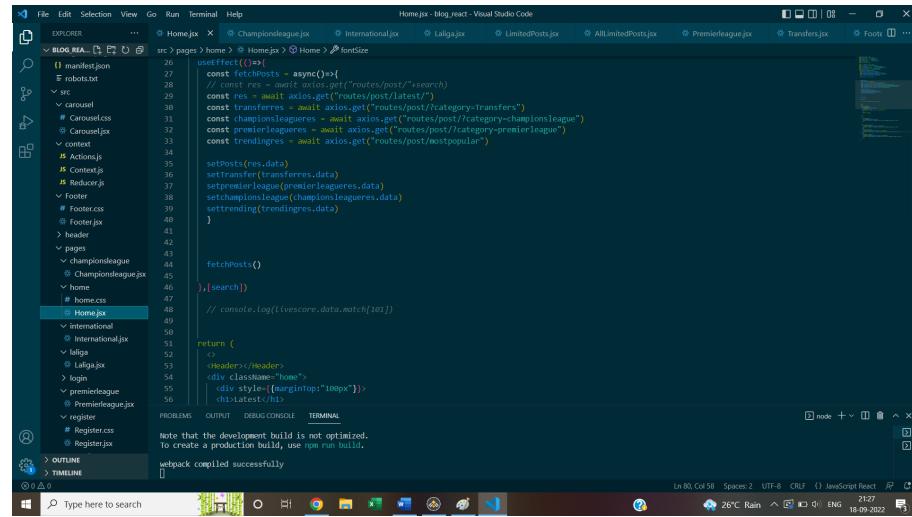


A screenshot of Visual Studio Code showing the code for retrieving blog posts. The code is located in the `post.js` file under the `routes` directory. It defines two routes: `/latest` and `/id`. The `/latest` route retrieves all posts from the database, sorts them by creation date, and returns them as JSON. The `/id` route retrieves a post by its ID and returns it as JSON.

```
router.get("/latest",async (req,res)=>{
  try{
    let posts;
    posts = await post.find({},null,{sort:[
      {"createdAt":1}
    ]})
    res.status(200).json(posts)
  }catch(err){
    res.status(500).json(err)
  }
})

router.get("/:id",async (req,res)=>{
  try{
    const posts = await post.findById(req.params.id);
    const {password,...others} = posts._doc;
    res.status(200).json(others)
  }catch(err){
    res.status(500).json(err)
  }
})
```

4.7 Code 3



The screenshot shows the Visual Studio Code interface with the file `Home.js` open. The code is a React component that fetches data from an API using Axios. It handles multiple categories of posts: latest, transfers, champion league, premier league, and most popular. The component uses functional state management with hooks like `useState` and `useEffect`. The code is well-organized with comments explaining the logic.

```
const fetchPosts = async()=>{
    const res = await axios.get("routes/post/latest")
    const transfers = await axios.get("routes/post?category=transfers")
    const championleagues = await axios.get("routes/post?category=championleague")
    const premierleague = await axios.get("routes/post?category=premierleague")
    const trendingres = await axios.get("routes/post/mostpopular")

    setPosts(res.data)
    gettransfers(transfers.data)
    setpremierleague(premierleague.data)
    setchampionleague(championleagues.data)
    settrending(trendingres.data)
}

fetchchests()
).{[search]}
// console.log(Livescore.data.match[10])
```

5 Test Cases

Table 6.1: Test Case - Login and Register

Test Case ID	Test Case Name	Test Data	Expected Output	Actual Output	Result
1	User enter user id and password	Enters the correct user id and password	Log in Successful	Home Page	Pass
2	User enter user id and password	Enters the user id and password	Prompt error	Prompt error	Pass
3	User enter user id and password	Valid user id and password which doesn't exist in Database	Registered Successfully	Login Page	Pass
4	User enter user id and password	Invalid user id and password which contains in Database	Prompt error	Prompt error	Pass

Table 6.2: Test Case - Others

Test Case ID	Test Case Name	Test Data	Expected Output	Actual Output	Result
1	User enters invalid type of post image file	Invalid image extension type	Prompt Message showing error	Successful Updated	Fail
2	User enters valid image type	Valid image upload	Image Update successful	Image Updated=successful	Pass

6 Limitations

- API for showing livescores doesnot have all types of match data .
- Blog doesnot have text formatting options
- Blog doesnot have option to upload multiple images in desired position in the posts.

7 Future Enhancements

- Use of Machine learning to recommend posts according to user's interest.
- Using a better version of API's to display more statistics related to matches.
- Using a better version of API to show more match scores
- Graphical representation of match statistics need to be shown

8 User Manual

Part 1 – Register

Upon opening the application, user will be greeted with the registration screen. If the user has no account, user can click on register and register self.

After verification of user id, User's account will be saved in our database. The user can now proceed to login.

Part 2 – Login

User needs to enter email first and then password. If there is an active internet connection, user can proceed to login.

Part 3 – Profile

User can access/add/update personal details (eg. Name, Emailid , Profile Photo). User can update this details anytime.

Part 4 – All Posts Section

User can read all posts available.

Part 5 – Posts Updates

User can update the posts they have written by clicking on the posts and clicking the update option

Part 6 – Write Posts

User can write their own posts by clicking on the write tab and then writing their own posts and clicking on the submit post button

9 Bibliography

9.1 Web References

- [1.] <http://www.youtube.com>
- [2.] <https://stackoverflow.com/>
- [3.] <https://www.draw.io/>
- [4.] <https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/>
- [5.] <https://www.geeksforgeeks.org/>