Date	Number Of Hours	Description of work
19/01/2025	3 Hours	The topic of the project, the languages to be used in it, the functioning of the project, the objective, and creating a niche project.
26/01/2025	½ hour	Project consultation gives the company an idea of what the project is about. And taking feedback on what work needs to be done and what has already been developed
26/01/2025	3 hours	Making a rough blueprint of the work, planning all the menu options, and backtracking seamlessly.
27/01/2025	7 hours	Coding the HTML, and CSS part of the project on Visual Studio, making use of GRIDS to help in making the website compatible with the screen ratios and making it responsive to even Mobile users
1/02/2025	6 hours	I was researching how to implement APIs for my project to give real-time news on a particular topic.
4/02/2025	1 hour	Setting up the repository, making sure the project is running successfully with the pushing and pulling of changes taking place Correctly
7/02/2025	3 hours	Remade the repository and solved all the errors I was getting; I was unable to deploy the project so I had to make a new repository.

14/02/2025-17/02-2025	2-3 hours daily	Working on the JavaScript part to make the selection of players work properly using drag and drop techniques.
20/02/2025	2 hours	Making the midterm project report , and recording the video about the presentation. Making Gantt charts to show the time spent to achieve a task.

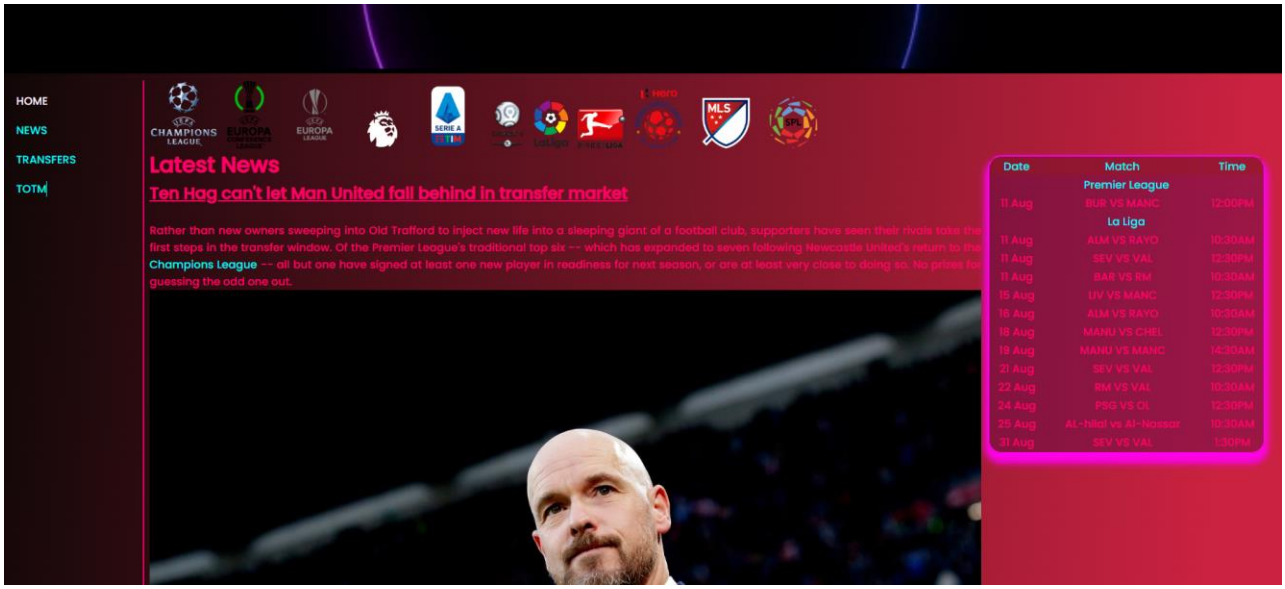
YOUTUBE VIDEO LINK: https://youtu.be/hKj_MddxNDc?si=4i82gzggu9XTQZRG
GITHUB REPOSITORY: https://github.com/Joshsans68/AppliedResearchRepo

Objective:

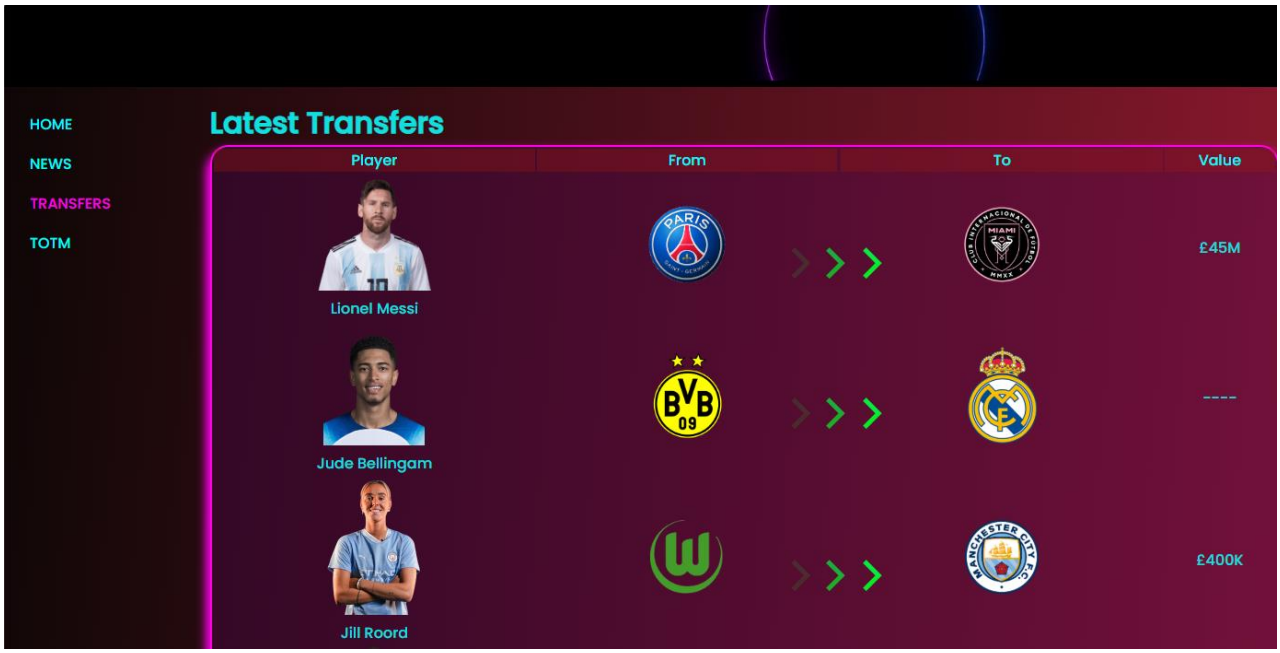
This report details the current state and future development plans for the "One Football" website, a platform designed to provide comprehensive football information and services.

Currently, the site consists of four key pages:

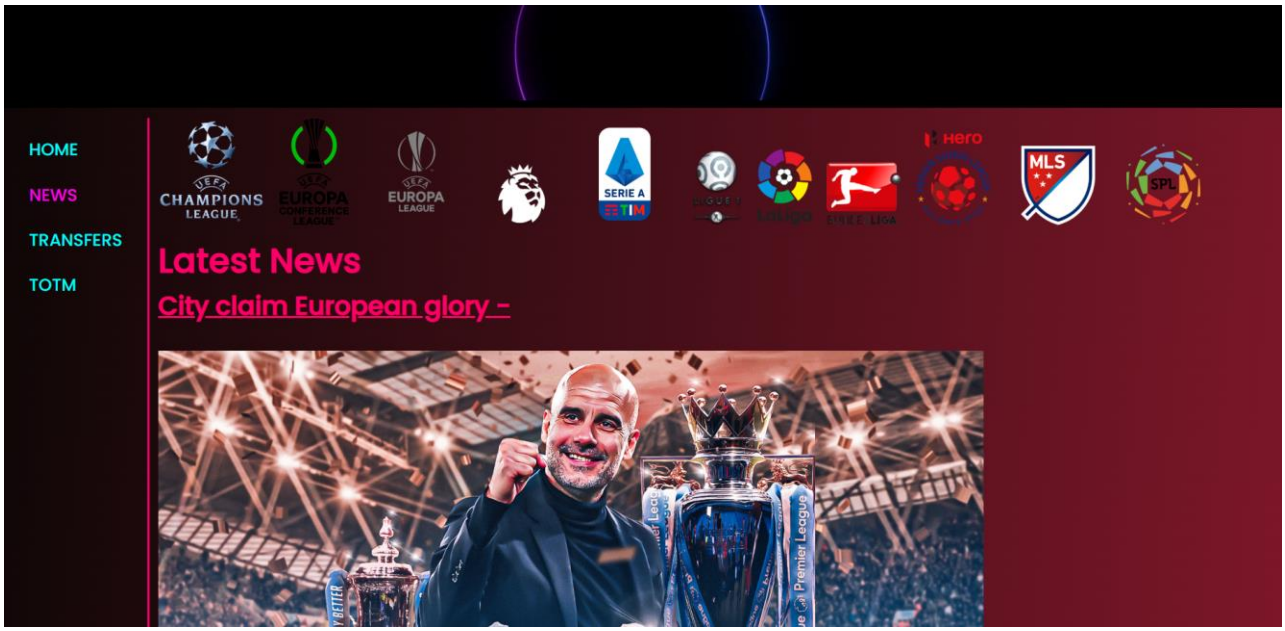
- **Landing Page:** The primary entry point for users, designed to showcase key features and attract engagement.



- **Transfers Page:** Dedicated to the latest football transfer news and rumors.



- **News Page:** Provides up-to-date football news from various sources.



- **Fantasy Teams Page:** A hub for information and links related to fantasy football platforms like bet365.

The goal is to enhance the website's functionality and user experience by incorporating API integrations for automated content updates and a ticketing system for stadium tickets.

Current Status

The existing website is built using HTML, CSS, and JavaScript. This foundation provides a solid base for future development. While the report doesn't have access to the specific implementation details, it assumes a standard structure where HTML provides the content and structure, CSS handles the styling, and JavaScript adds interactivity.

Strengths:

Clear Structure: The four-page structure offers a logical organization of information.

Technology Stack: The chosen technologies (HTML, CSS, JavaScript) are widely used and well-supported, providing a good starting point.

Areas for Improvement:

Manual Content Updates: Currently, content updates are likely manual, which is time-consuming and prone to delays.

Lack of Dynamic Content: The site likely lacks dynamic content updates, limiting user engagement.

No Ticketing System: The absence of a ticketing system prevents users from directly

purchasing tickets through the platform.

Future Development

The primary focus of future development is to integrate APIs for automation and implement a ticketing system. Node.js will be used on the backend to facilitate these integrations.

1. API Integration for Automation:

News API: Integrating a news API will automate fetching and displaying football news. This will ensure that the news page is always up to date with minimal manual intervention. Consider APIs like ESPN API, News API, or others specializing in sports news.

Transfer News API: A dedicated transfer news API will automate updates on player transfers, rumors, and deals. This will enhance the value of the Transfers page.

Fantasy Data API: Integrating an API for fantasy football data (if available for platforms like bet365) could provide real-time updates on player performance, team standings, and other relevant information. This could significantly enrich the Fantasy Teams page.

Implementation Details (API Integration):

Node.js Backend: A Node.js server will handle API requests, process the data, and serve it to the front end.

Data Storage: Depending on the volume and frequency of API calls, a database (e.g.,PHPmyADMIN) will be necessary to cache the data and improve performance.

Frontend Updates: JavaScript will be used to fetch data from the Node.js server and dynamically update the content on the respective pages.

2. Ticketing System Integration:

Ticketing API: Integrating with a ticketing platform's API (e.g., Ticketmaster, StubHub, or a dedicated football ticketing service) is crucial. This will allow users to browse available tickets, select seats, and make purchases directly through the

"One Football" website.

User Authentication: A user authentication system will be required to manage user accounts and ticket purchases.

Payment Gateway Integration: Integrating a payment gateway (e.g., Stripe, PayPal) will enable secure online transactions.

Implementation Details (Ticketing System):

Node.js Backend: The Node.js server will handle communication with the ticketing API, manage user accounts, and process payments.

Database: A database will be essential to store user data, ticket information, and order history.

Frontend Development: New UI elements will be developed to display available tickets, manage bookings, and handle payment processing.

Technology Stack (Revised)

Frontend: HTML, CSS, JavaScript
Backend: Node.js

Database (Optional/Likely): MongoDB, PostgreSQL, or similar

API Integrations:

News API, Transfer News API, Fantasy Data API (if available), Ticketing API, Payment Gateway API

Development Roadmap:

Backend Development (Node.js): Set up the Node.js server, implement API integration logic, and design database schema (if needed).

Frontend Enhancements: Update the frontend to fetch data from the backend and dynamically update the content. Develop new UI components for the ticketing system.

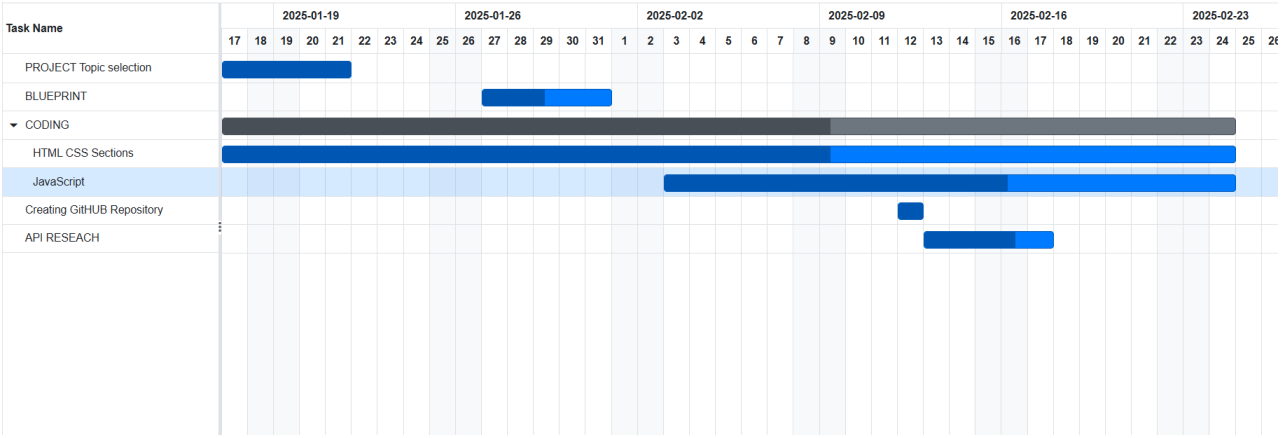
API Integration: Integrate with the chosen news, transfer, fantasy (if applicable), and ticketing APIs.

Testing: Thoroughly test the website to ensure functionality, performance, and security.
Deployment: Deploy the website to a hosting platform.

Conclusion:

The "One Football" website has the potential to become a valuable resource for football fans. By integrating APIs for automation and implementing a ticketing system, the website can offer a more dynamic, engaging, and comprehensive user experience. The proposed technology stack, combined with a well-defined development roadmap, will ensure the successful implementation of these enhancement.

Gantt Chart:



This report summarizes the progress of the project based on the Gantt chart provided, covering the period from January 19th, 2025, to February 23rd, 2025.

1. Project Topic Selection (Completed):

- **Start Date:** January 19th, 2025
- **End Date:** January 23rd, 2025
- **Status:** Complete

The initial phase of selecting the project topic has been successfully concluded. This foundational step has established the project's focus and scope, guiding subsequent development efforts.

2. Blueprint (Completed):

- **Start Date:** January 26th, 2025
- **End Date:** January 30th, 2025
- **Status:** Complete

A comprehensive blueprint for the project has been developed. This likely involved defining project goals, outlining requirements, creating design or architecture, and establishing a project timeline and strategy. This blueprint serves as a roadmap for the development process.

3. Coding (In Progress):

- **Start Date:** January 23rd, 2025
- **End Date:** February 23rd, 2025
- **Status:** In Progress

The **coding phase,** encompassing the actual development of the project, is currently underway. This is the most extensive phase and is broken down into specific sub-tasks.

The **development of the HTML** structure and **CSS styling** for the project has been finalized. This establishes the user interface's layout, design, and basic structure.

The implementation of **JavaScript functionality** for the project has been completed. This enables dynamic and interactive elements within the application.

4. Creating GitHub Repository (Completed):

- **Start Date:** February 9th, 2025
- **End Date:** February 11th, 2025
- **Status:** Complete

A GitHub repository has been successfully created. This establishes version control for the project, facilitating collaboration, tracking changes, and managing the codebase effectively.

5. API Research (Completed):

- **Start Date:** February 9th, 2025
- **End Date:** February 18th, 2025
- **Status:** Complete

Research into potential APIs for enhancing the project's functionality has been completed. This likely involved identifying suitable APIs, evaluating their capabilities, and gathering necessary information for integration.

Summary and Next Steps:

The project has made significant progress, with the planning, design, and core front-end development (HTML, CSS, and JavaScript) phases completed. Version control is in place via GitHub, and API research has been concluded.

The next step is to finalize the coding phase, scheduled for completion on February 23rd, 2025. This will likely involve integrating the researched APIs, implementing any necessary backend logic, and addressing any remaining development tasks. Following the completion of coding, testing, and deployment phases would be expected, though they are not represented in the provided Gantt chart.

Repo Check-In:

Since the last progress report, several key implementations have been checked into the repository. The initial setup of the project structure, including defining the objective and scope, was completed. A rough blueprint of the project, including menu planning and navigation flow, was designed and documented. The HTML and CSS components were developed using Visual Studio, incorporating GRIDS for responsive design to ensure compatibility across different screen sizes, including mobile devices. Additionally, API research was conducted to integrate real-time news updates into the project. The final repository setup was also completed, ensuring the smooth pushing and pulling of code changes for seamless collaboration.

As of 24-Feb-25
Things are moving along on the project. We started by nailing down the topic and then created a blueprint for the overall structure. From there, we built the core front-end: HTML for the structure, CSS for the styling, and then added JavaScript for interactivity. All that front-end work is done.
We also got the GitHub repository set up for version control and collaboration. At the same time, we were researching APIs to pull in real-time data – that research phase is also complete.
The main coding phase is still ongoing and scheduled to wrap up around February 23rd. That's where we're focusing our efforts now – integrating the APIs and putting everything together.