Final Report

Paris 2024 Olympics Data Visualization Web Application

Video Demonstration:

https://drive.google.com/file/d/1MfxpW6IvnVMch3Zi5HD5foTpKKHeynbF/view?usp=sharing

Course Code: CSE 327

Course Name: Software Engineering

Section: 06

Semester: Summer 2024

Prepared by: Group 08

- Ahsan Rizvi (2122272642)
- Mohammad Irtiza Hossain Mahmud (2014321642)
- Afsana Umme Kulsum Eaty (1922125642)

Submitted to:

Dr. Mohammad Rezwanul Huq [MRH1]
Associate Professor (Part-time), North South University

Table of Contents

1. Software Requirement Specification (SRS)

1.1 Purpose of the Document

The purpose of this Software Requirements Specification (SRS) document is to provide a comprehensive understanding of the Paris 2024 Olympics Data Visualization Web Application, designed to offer interactive tools for users to explore Olympic datasets related to the upcoming games.

1.2 Scope of the Document

This document outlines the system requirements and features necessary for the development and deployment of the Paris 2024 Olympics Data Visualization Web Application. It describes the key functionalities, including data filtering, visualization, and export options, aiming to provide a seamless and engaging experience for users interested in the Olympics.

1.3 Overview

The goal of the Paris 2024 Olympics Data Visualization Web Application is to offer a one-stop platform where users can explore, analyze, and visualize Olympic datasets. Features include filtering data by country, sport, athlete, and medal standings, along with tools for graphical representation and data export.

1.4 Intended Audience

The intended audience for this web application includes:

- Sports enthusiasts and fans
- Journalists and analysts
- Researchers
- Paris 2024 Olympics organizing committee

1.5 Intended Use

The web application will be used to:

- Access detailed Olympic event data
- Explore datasets by sport, country, or athlete
- Visualize data through various graphs and charts
- Export data in CSV or PDF formats for offline analysis

1.6 Definitions and Acronym

- **Data Visualization:** A graphical representation of data for easier understanding.
- **CSV:** Comma-Separated Values, a format for tabular data.
- **Olympics:** International multi-sport event featuring summer and winter sports competitions.

2. Overall Description

2.1 User Needs

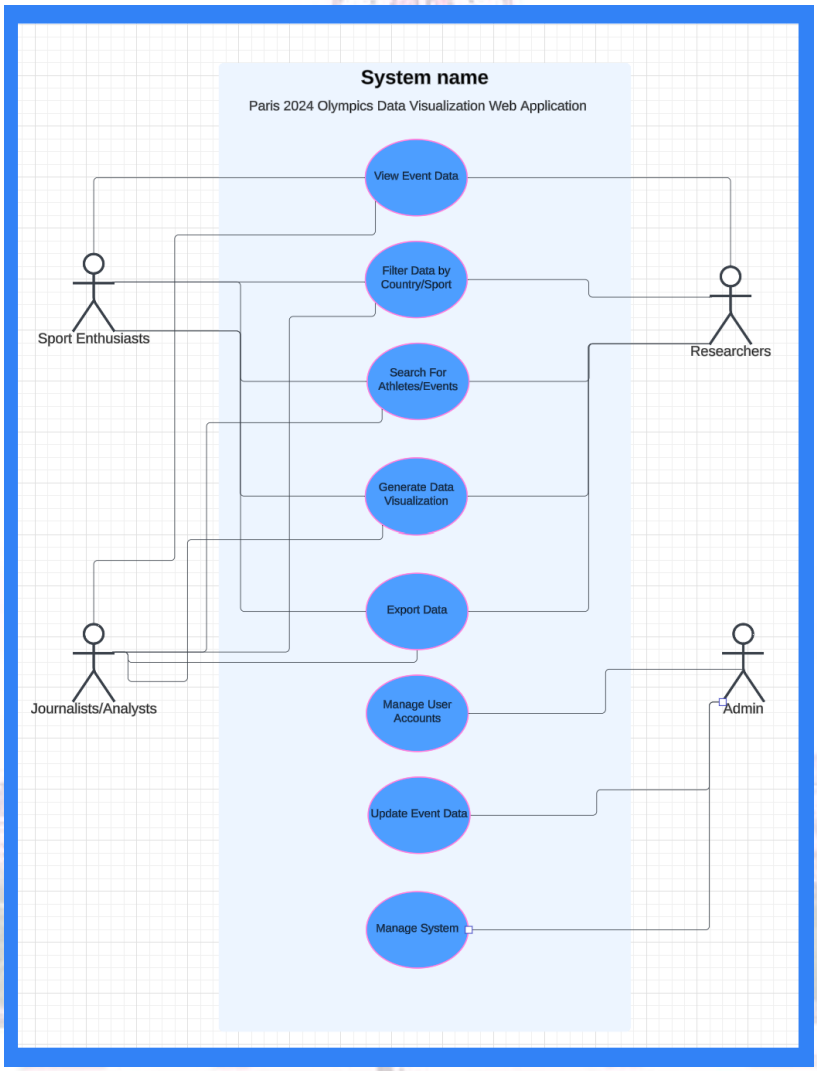
The web application will meet the following user needs:

- **Sports Enthusiasts:** Ability to view real-time data about Olympic sports, events, and athletes.
- **Journalists and Analysts:** Access to comprehensive datasets and visualizations for reporting.
- **Researchers:** Tools for detailed data exploration and export for in-depth analysis.

2.2 Assumptions and Dependencies

The web application will be developed using modern web technologies, accessible on any major web browser (e.g., Chrome, Firefox). A stable internet connection will be required for real-time data access.

2.3 Use Case Diagram



3. System Features and Requirements

3.1 Functional Requirements

- **User Authentication:** Users can create accounts, log in, and manage their profiles for saved preferences and data exports.
- **Data Display:** Users can view data for Olympic events, athletes, and medal standings.
- **Data Filtering:** The system allows users to filter data by country, sport, event, and athlete.
- **Search Functionality:** Users can search for specific events, athletes, or countries using a dynamic search bar.
- **Data Visualization:** The system offers multiple data visualization formats, including bar graphs, pie charts, and line graphs.
- **Data Export:** Users can export filtered data in CSV and PDF formats.

3.2 External Interface Requirements

- **User Interface:** A web-based interface accessible on both desktop and mobile browsers.
- **APIs:** External APIs will be used to fetch real-time Olympic event data and athlete statistics.
- **Database:** A backend database will store and manage Olympic data and user-generated reports.

3.3 System Features

- **User Registration and Login:** Users can register and log into the application, managing their profiles and saving preferences.
- **Event and Athlete Information:** Detailed data on Olympic events, athletes, and medal standings.
- **Filtering and Search:** Ability to filter and search for specific datasets.
- **Data Visualization:** Users can generate visualizations for better understanding of data.
- **Data Export:** Users can download data in CSV or PDF formats.

3.4 Business Rules

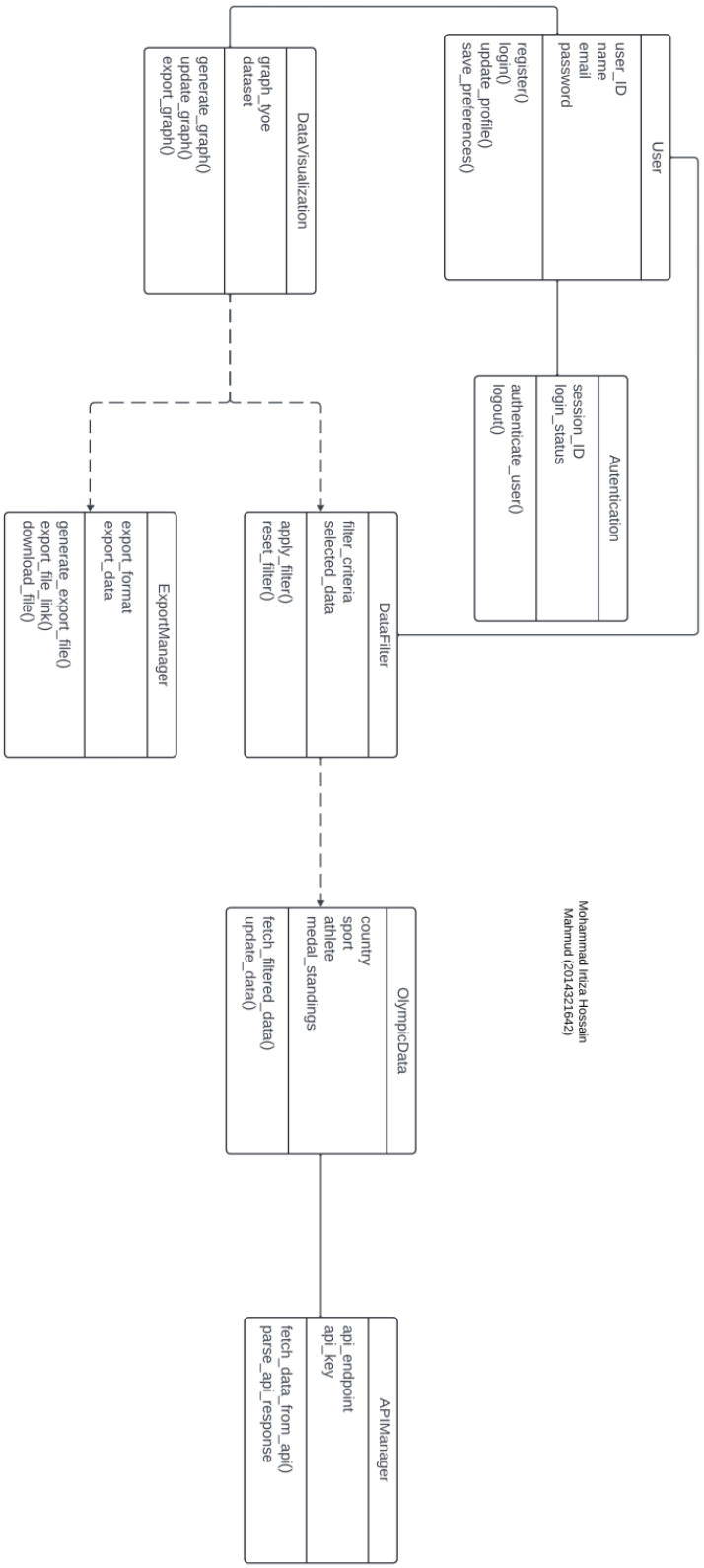
- **Data Accuracy:** The displayed Olympic data must be accurate and up-to-date.
- **Fair Usage:** Users must not attempt to manipulate or misrepresent data.
- **User Account Rules:** Users must provide valid information during account creation.

3.5 Non-Functional Requirements

- **Performance:** The system must respond to user actions within 2-3 seconds.
- **Usability:** The interface must be intuitive and easy to navigate, with consistent design elements.
- **Security:** User authentication will be secured, and data transfers will use encryption protocols.
- **Reliability:** The system must ensure 99% uptime during peak usage periods.
- **Device Compatibility:** The system will support responsive design to adapt to different screen sizes on desktop and mobile.
- **Portability:** The application must be responsive and work across all devices, including desktops, laptops, tablets, and smartphones.

2. System Modeling Diagrams

1. Class Diagram

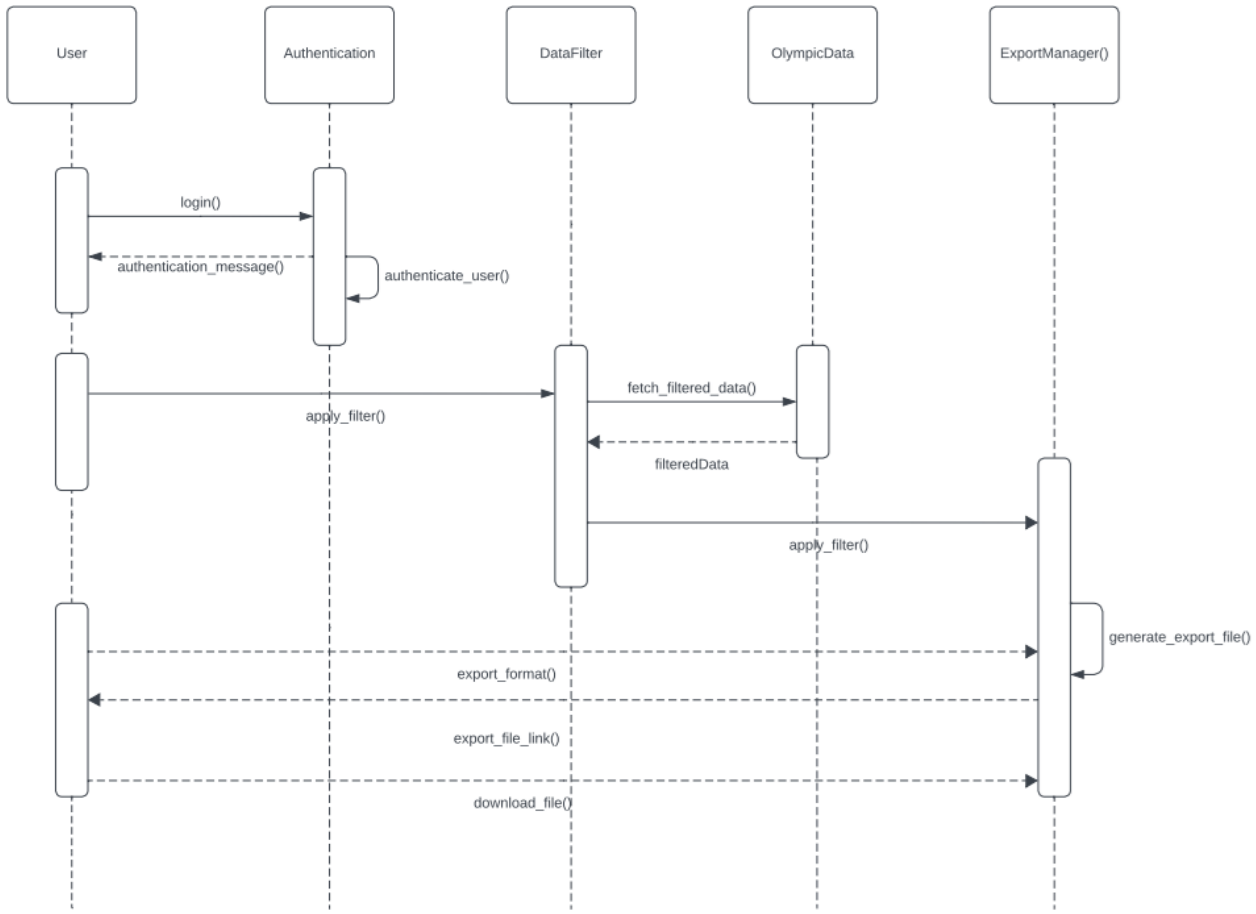


Mohammad Iriza Hossain
Mahmud (2014321642)

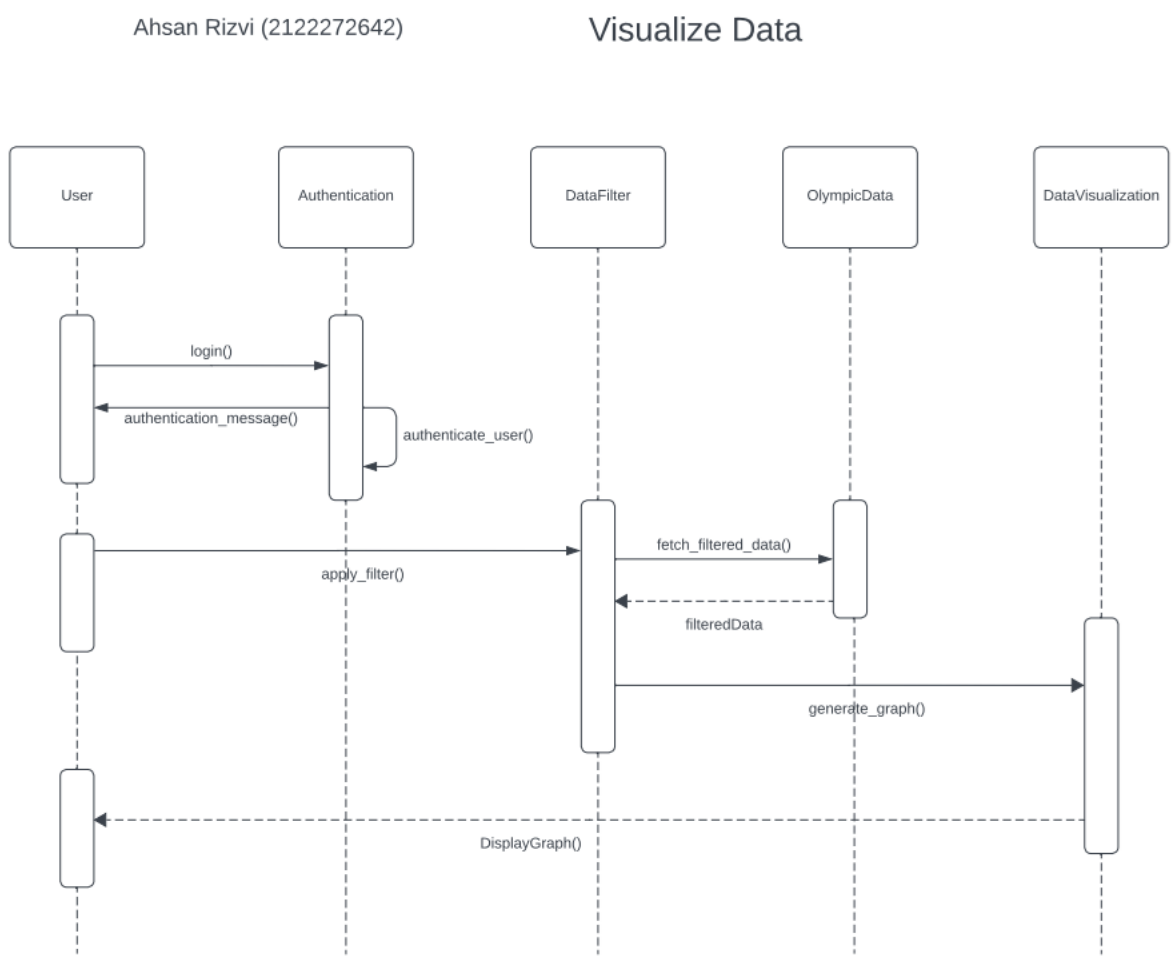
2.1. Sequence Diagrams

Ahsan Rizvi (2122272642)

Export Data

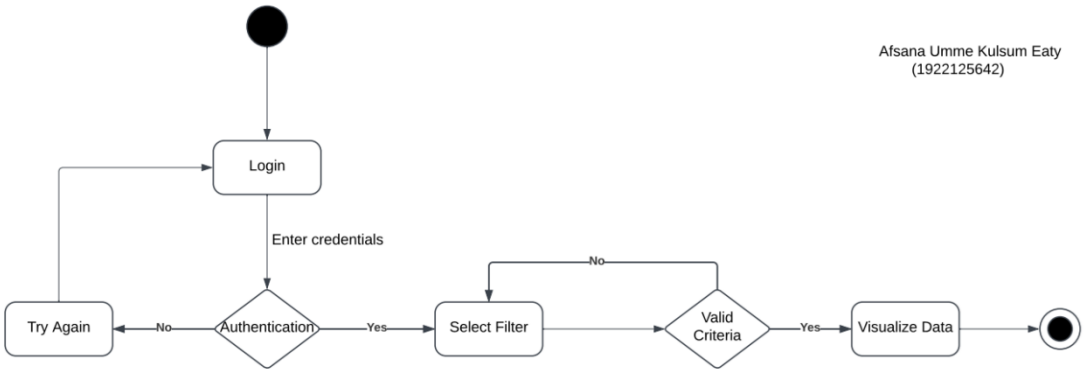


2.2. Sequence Diagrams

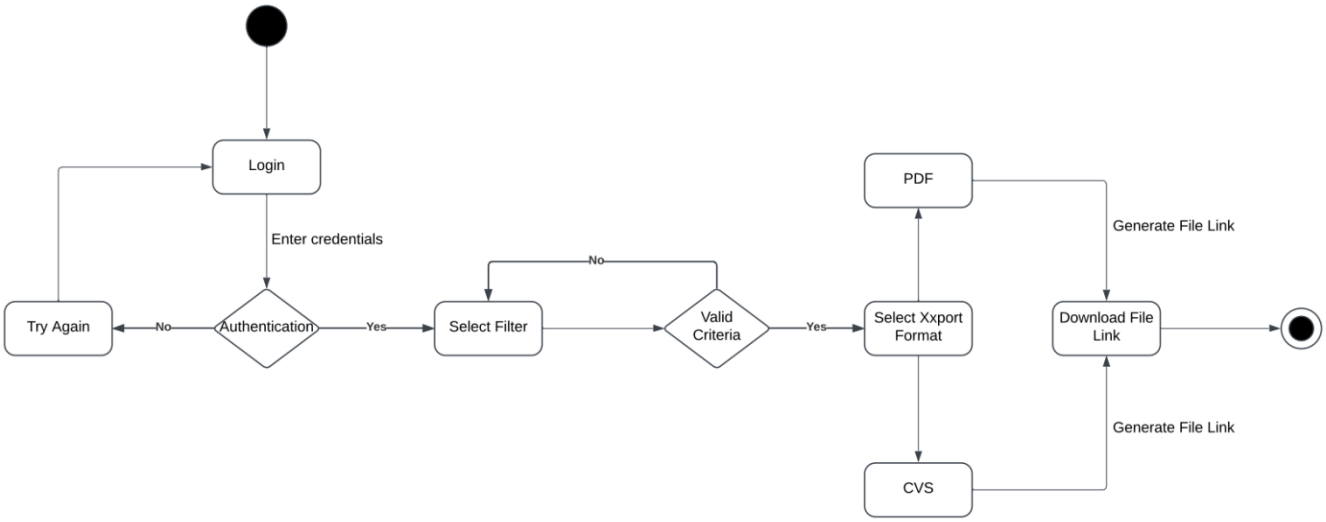


3. Activity Diagrams

Visualize Data

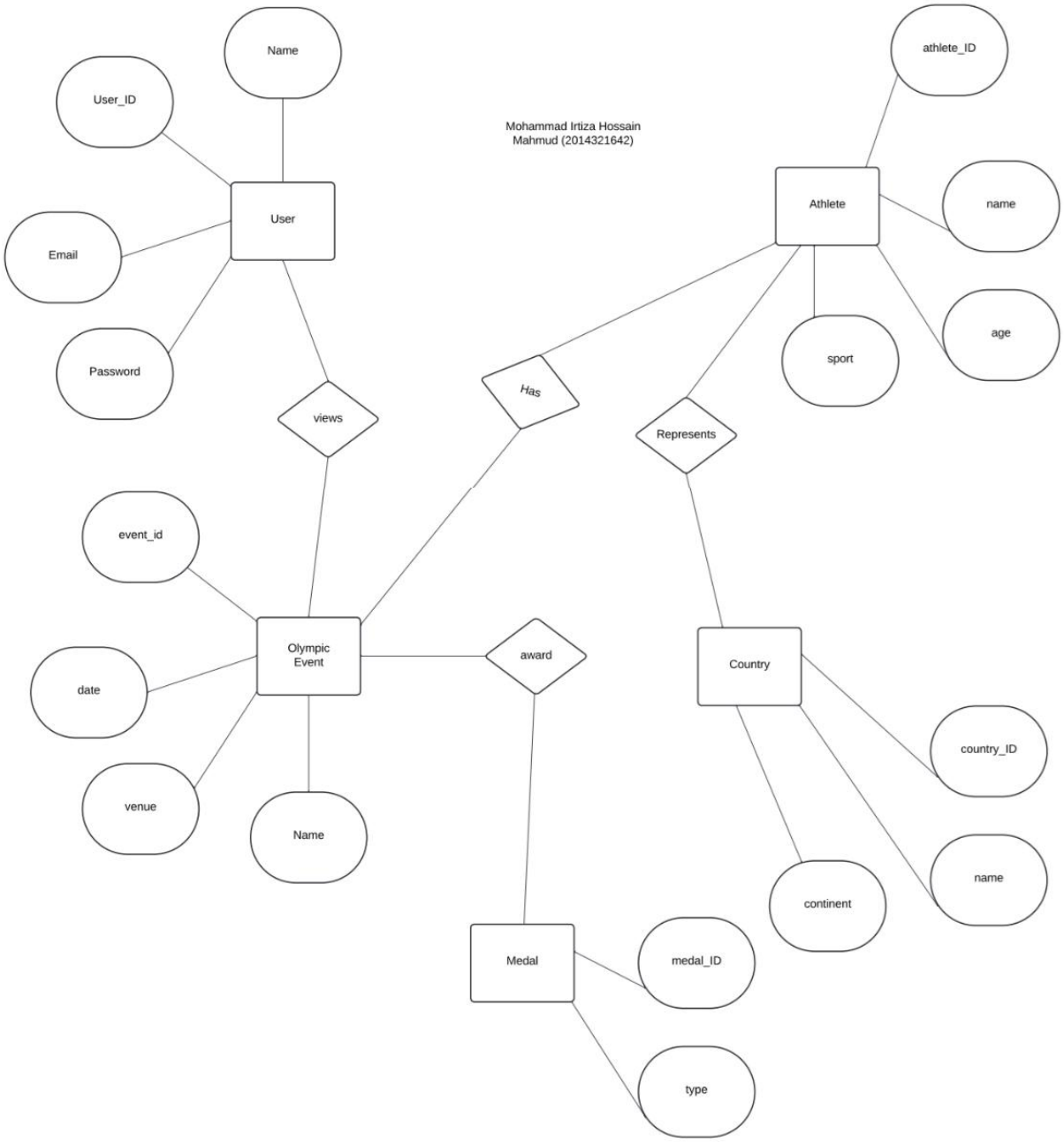


Export Data



3. Database Design

ER Model:



4. User Access Control

The Paris Olympics 2024 web application integrates a user authentication system built using Django's built-in authentication framework. The system defines three types of user access: Admin users, authenticated users, and unauthenticated users. Each user type has a different set of permissions and access levels within the application.

Admin Users:

- Admin users have unrestricted access to all parts of the web application.
- They can manage athletes, events, and medals via the Django admin interface, which is accessible through /admin.
- Admin users can export any data (athletes, events, and medals) in both CSV and PDF formats, without any filtering limitations.
- They have full control over the user base, allowing them to create, update, or delete user accounts.

Authenticated Users:

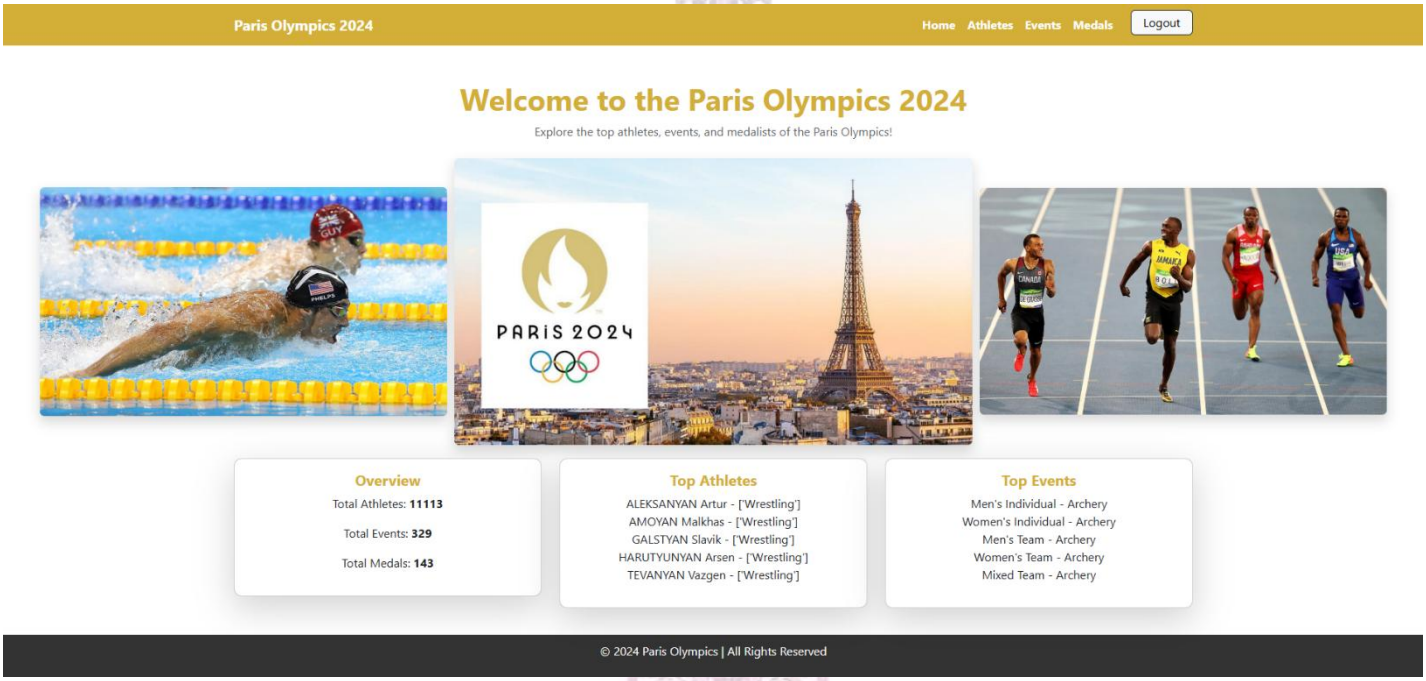
- Authenticated users can access detailed pages for athletes, events, and medals after logging in.
- They can filter data based on specific search criteria such as athlete name, sport, event, and medal type.
- They can visualize data in graphical formats such as bar charts and pie charts.
- Authenticated users are allowed to export filtered data (athletes, events, or medals) as CSV or PDF files, based on their search and filter criteria.

Unauthenticated Users:

- Unauthenticated users can only access the login and registration pages.
- They cannot view the athletes, events, or medals pages or interact with the data in any way.
- They are redirected to the login page when attempting to access any restricted content.
- This role-based access control ensures that the web application maintains data integrity and security while providing a streamlined user experience for both the admin and authenticated users.

5. Functionality-wise System Description

1. Home Page



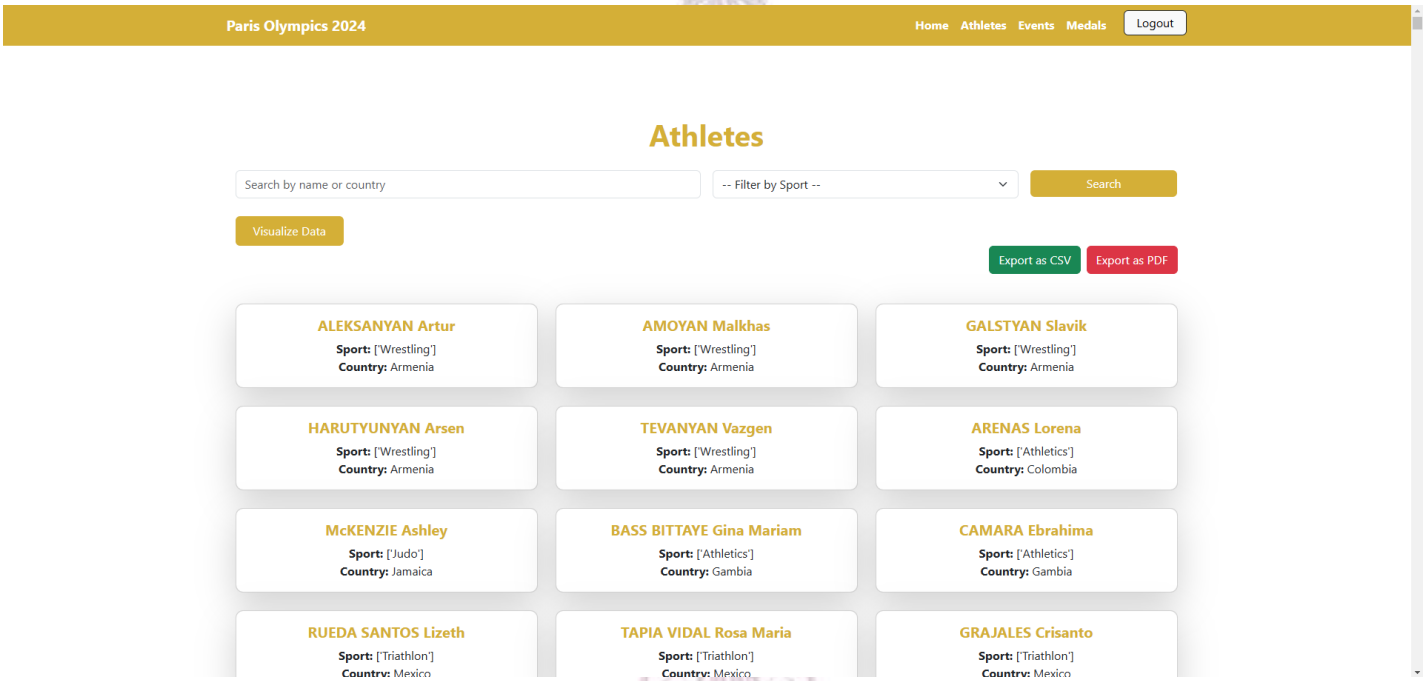
Overview:

The home page provides a summary of the Olympics data, including the total number of athletes, events, and medals. It also highlights the top athletes and events.

Features:

- Displays total statistics (total athletes, events, and medals).
- Lists the top athletes and top events with the sport type.
- Includes an aesthetically pleasing layout with images representing the Olympic Games.

2. Athletes Page



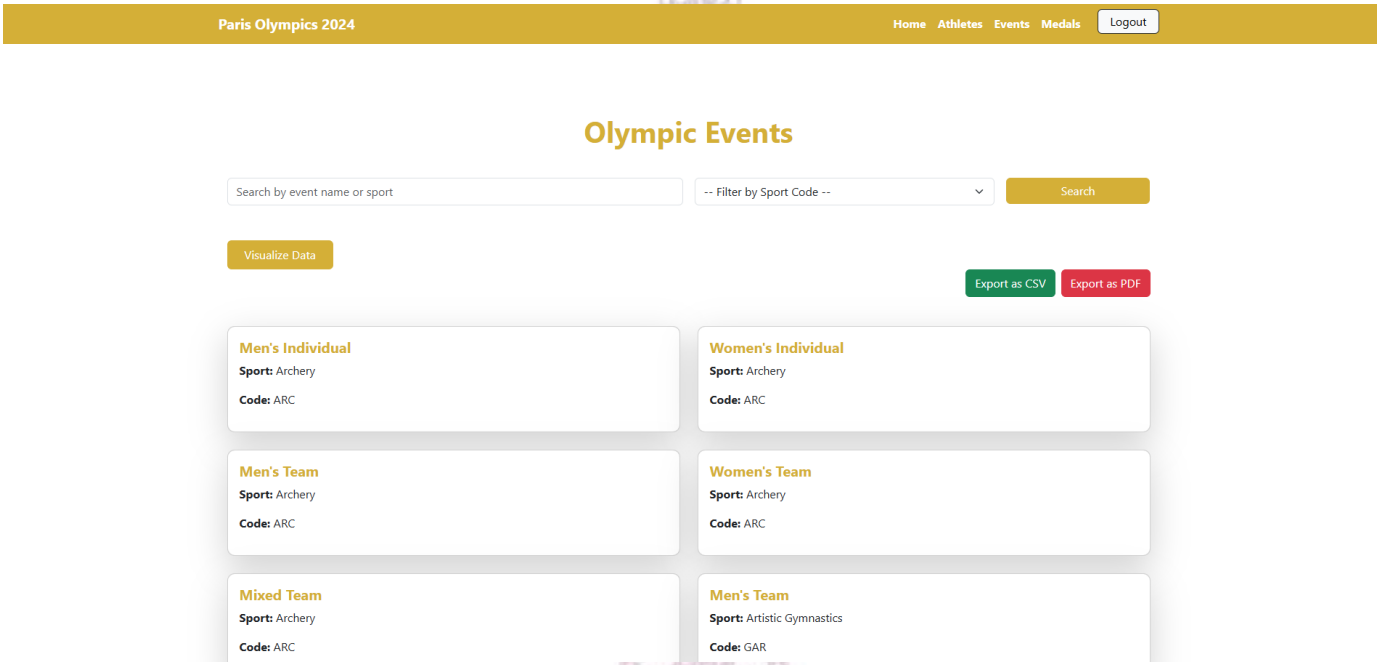
Overview:

The athletes page displays a list of athletes, allowing users to search and filter athletes by name, country, or sport. It also provides the option to visualize the athlete data by country using bar charts.

Features:

- Search by athlete name or country.
- Filter athletes by sport.
- Export filtered athlete data to CSV or PDF.
- Visualize athlete data in a bar chart that shows the number of athletes per country.

3. Events Page



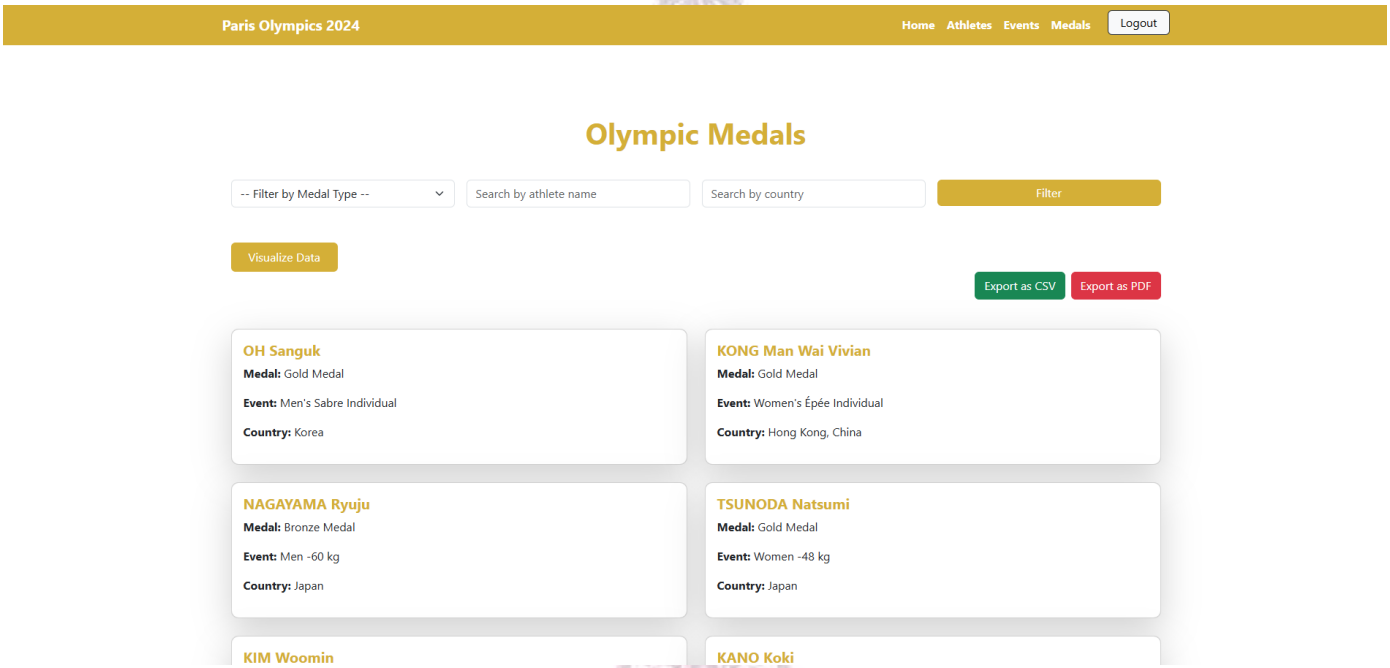
Overview:

The events page lists all Olympic events, with search and filter options for sport codes and event names. The page also provides data visualization of events by sport in a pie chart format.

Features:

- Search events by name or sport.
- Filter by sport code.
- Export filtered event data to CSV or PDF.
- Visualize event data by sport, displayed in a pie chart format.

4. Medals Page



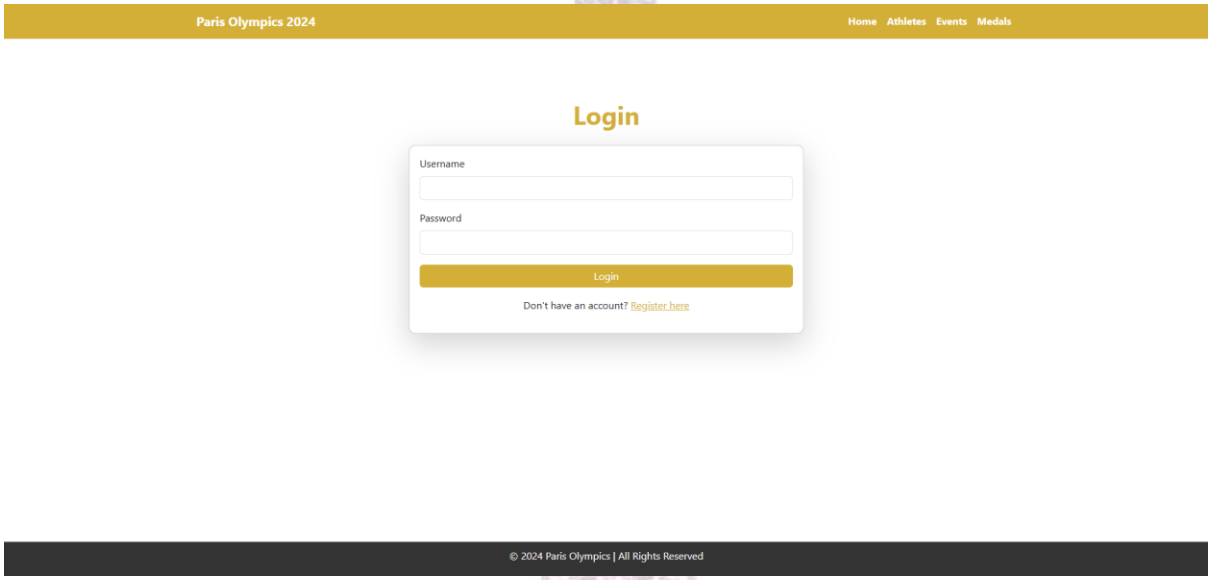
Overview:

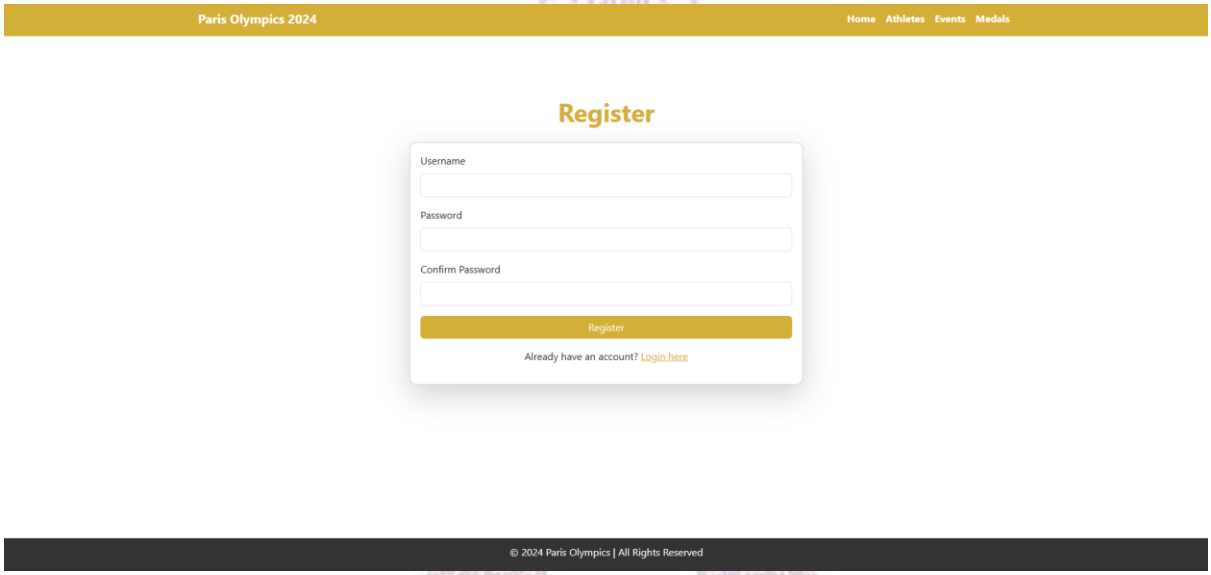
The medals page shows a list of medals won, including details such as athlete names, countries, events, and medal types. Users can filter by medal type, athlete name, and country, with an option to visualize medal statistics by country.

Features:

- Search by athlete name, country, or medal type.
- Export filtered medal data to CSV or PDF.
- Visualize medals data by country, displayed in a bar chart.
- Screenshot: (Insert screenshot of the Medals Page here)

4. Login and Registration Pages





Overview:

Users must log in to access the protected pages of the site (athletes, events, medals). The login page provides a form for authentication, while the registration page allows users to create an account.

Features:

- Login form with error handling for invalid credentials.
- Registration form that allows users to create an account by providing a username and password.

6. Project Scheduling using Critical Path Method (CPM)

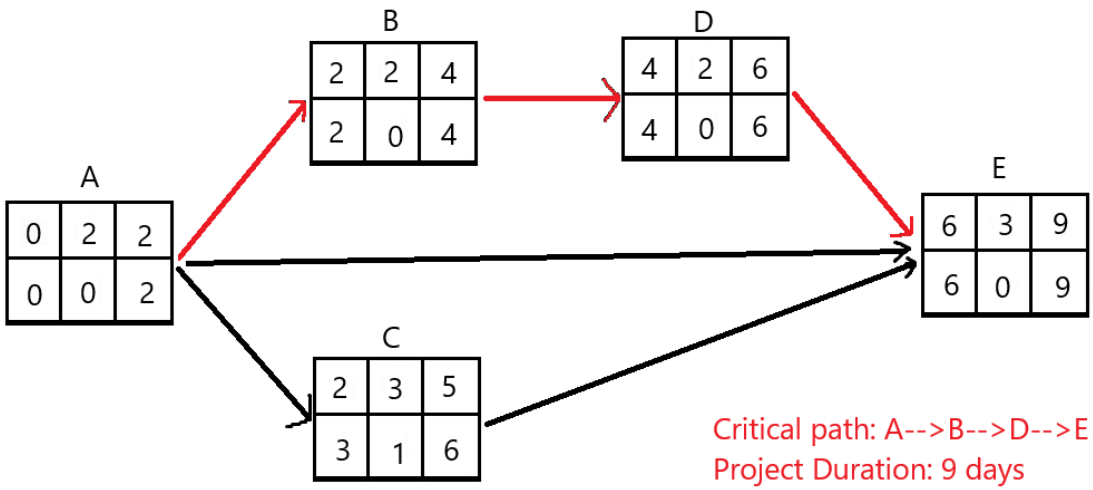
To efficiently plan and execute the Paris Olympics 2024 web application, we can apply the Critical Path Method (CPM). CPM helps identify the sequence of tasks that are critical to the project's success and ensures timely delivery. Below is a detailed project schedule, broken down by tasks, dependencies, and estimated durations.

Task Identification

A. **Data Models:** Create models for athletes, events, and medals.
B. **User Authentication**: Implement login, registration, and logout features.
C. **Frontend Design**: Design and implement HTML templates for displaying athletes, events, and medals.
D. **Export Functionality**: Implement CSV and PDF export for filtered data.
E. **Visualization**: Implement data visualization for athletes, events, and medals.

Project Timeline

Task	Duration	Dependencies
Task A: Models Creation	2 days	None
Task B: User Authentication	2 days	Task A (Models Creation)
Task C: Frontend Design	3 days	Task A (Models Creation)
Task D: Data Export	2 days	Task B (User Authentication)
Task E: Visualization	3 days	Task A (Models Creation), Task C (Frontend Design), Task D (Data Export)



7. Modeling and Implementation Challenges

Modeling Challenges:

While working on the data modeling for the Paris Olympics 2024 web application, we encountered several challenges, primarily due to incomplete or missing data in the original dataset. The models were created based on the relationships between Athletes, Events, and Medals, with ForeignKey relationships to ensure data integrity. However, there were several issues:

1. **Incomplete Data:**

Many attributes of the models, such as birth date, birth place, height, weight, and coach for the Athletes model, were missing or not consistently provided in the dataset. As a result, we could not fully populate the database for these attributes, and had to skip many of them during the import process.

To mitigate this, we focused on the name, sport, and country attributes, which were always available. We prioritized these key attributes and left the other optional ones empty, using Django's ability to handle null=True and blank=True for optional fields.

2. **Data Integrity:**

When importing Medals and Events, there were issues with referencing athletes and events that did not exist in the dataset. In some cases, certain medals and events lacked corresponding data, so we had to filter out or handle missing values.

While importing Medals, we encountered some records where the Athlete or Event was missing from the database, which led to skipped entries. This issue was mitigated by adding validation checks during the import process, ensuring that only complete data was inserted into the database.

Implementation Challenges

After addressing the data modeling challenges, we moved forward with implementing the application's functionality. This phase introduced several challenges that required multiple adjustments to the system:

1. **Data Visualizations:**

While generating the bar graph for Athletes by Country, we faced an issue with the sheer volume of data. Displaying all countries on the graph made it overly cluttered and difficult to interpret. To resolve this, we limited the graph to display only the top 10 countries by the number of athletes.

Similarly, when visualizing Events and Medals, the data became overwhelming due to the large number of options. To maintain clarity, we restricted the graphs to show the top 10 to 15 entries, ensuring that the visualizations were both meaningful and manageable.

2. Login, Logout, and Registration Redirecting Issues:

Initially, we faced problems with proper redirection after users logged in, logged out, or registered. Users were sometimes not redirected to the appropriate pages, which caused confusion.

After several adjustments to the view functions and login logic, these issues were resolved by properly handling redirects. The use of Django's built-in login and logout views ensured the seamless navigation of users through the authentication process.

3. Export Functionality (CSV and PDF):

Exporting filtered data (CSV and PDF) posed another significant challenge. Initially, the export functionality would export all data, regardless of the filters applied on the page. This was fixed by passing the filter parameters in the export URLs, ensuring that only the data matching the applied filters was exported.

Additionally, handling the PDF generation involved ensuring that the correct filtered data appeared in the exported document. This required careful manipulation of data in the views and custom template rendering for the PDF export to work as expected.

4. Frontend Design and Layout Issues:

A significant portion of time was spent addressing frontend design challenges. Issues such as improper spacing, text not displaying correctly, and layout misalignments arose frequently. These issues were particularly noticeable when designing the search forms and ensuring that results and buttons were displayed properly.

Careful attention was given to ensure proper styling and layout adjustments. Using Bootstrap's grid system and customizing the HTML and CSS made sure that the pages were responsive, visually appealing, and aligned correctly.

Another challenge in the frontend was ensuring that each page was coherent with its corresponding functionality in views.py. The templates had to be adjusted frequently to reflect the data passed from the views, ensuring consistency between the user interface and the backend logic.

8. Conclusion

The Paris Olympics 2024 web application successfully meets the goal of providing a comprehensive system to manage and visualize Olympic data. The following features have been successfully implemented:

- **Search and Filter**: Users can search and filter athletes, events, and medals.
- **Data Visualization**: Provides interactive bar charts and pie charts for visualizing athlete data by country and event data by sport.
- **Exporting Data**: Allows authenticated users to export filtered data as CSV or PDF.
- **User Authentication**: Ensures secure access to the application with role-based access control.

While there were challenges related to user authentication, data export, and visualization, these were effectively overcome using Django's robust features and libraries like matplotlib and xhtml2pdf.

This project has provided a comprehensive learning experience in web development with Django, focusing on dynamic data handling, authentication, and creating rich user interactions through data visualization and exports. The system can easily be extended to support more complex features, such as real-time data updates or user contributions.

9. Contributions

Name:	Contributions:	Signature:
Ahsan Rizvi (2122272642)	**Phase 01 (SRS):** • Did 1.1 to 1.6 • Prepared the report **Phase 02 (UML):** • Sequence Diagrams **Phase 03:** • Entire Backend • User Access Control • Modeling and Implementation Challenges • Conclusion	
Mohammad Irtiza Hossain Mahmud (2014321642)	**Phase 01 (SRS):** • Did 2.1 to 2.3 in SRS • Use Case Diagram **Phase 02 (UML):** • Class Diagram • ER Diagram **Phase 03:** • Frontend except Homepage • Functionality-wise System Description	
Afsana Umme Kulsum Eaty (1922125642)	**Phase 01 (SRS):** • Did 3.1 to 3.5 in SRS **Phase 02 (UML):** • Activity Diagrams • Prepared the report **Phase 03:** • Frontend - Homepage • Project Scheduling • Prepared the Report