SPL-2 Project Report

Mapping The World

Submitted by

1) Rupali Tasnim Samad
Roll : 1208
Session : 2019-20

2) Tashfia Jannath
Roll : 1223
Session : 2019-20

Supervised by

Moumita Asad

Lecturer

Institute of Information Technology,

University of Dhaka

Institute of Information Technology

University of Dhaka

29-05-2023

SIGNATURE PAGE

Project : Mapping The World

Author : Rupali Tasnim Samad
(Roll - 1208)

Tashfia Jannath
(Roll - 1223)

Date Submitted : May 29th, 2023

Supervised By : Moumita Asad
lecturer
Institute of Information Technology,
University of Dhaka.

Supervisor's Approval :

____

_____

(Signature of Moumita Asad)

Index of Tables

Index of Figures

CHAPTER 1

INTRODUCTION OF MAPPING THE WORLD

Mapping The World is an innovative website that serves as a treasure trove of knowledge for country enthusiasts and an invaluable tool for students. Leveraging modern technology, this platform offers a multitude of features to create an immersive and interactive learning experience. With its real-time chatting system, users can engage in conversations with like-minded individuals from around the globe, fostering cultural exchange and broadening perspectives. The quiz section challenges users to test their knowledge and reinforces learning outcomes. Furthermore, the question-answer forum provides a platform for users to seek and share information, promoting a collaborative environment. Whether satisfying the thirst for knowledge or assisting students in their studies, Mapping The World is a comprehensive resource that harnesses the power of technology to create a global educational community.

1.1 Motivation

The motivation behind the development of MappingTheWorld stems from the recognition of several critical factors. Firstly, there exists a lack of engaging and interactive learning resources for individuals to explore and understand global geography, history, and related topics. Traditional learning methods often fail to capture the attention and interest of students, leading to a disengagement with these subjects. Additionally, there is a prevailing stereotype that places a higher value on science subjects, leading to a decreased emphasis on humanities education.

The project aims to address these challenges by providing an innovative and entertaining learning platform that breaks the monotony of traditional education. By leveraging modern technology and interactive features, MappingTheWorld seeks to create a stimulating learning environment where individuals of all ages can explore and discover the fascinating aspects of the world. By presenting information in an interactive manner, the project aims to facilitate better retention of knowledge, making it easier for students to memorize and understand the facts.

One of the primary motivations behind MappingTheWorld is to challenge the existing stereotype that prioritizes science over humanities. By introducing a dynamic and captivating learning experience in the field of geography, history, and humanities, the project aims to shatter the notion that studying these subjects is inferior. By igniting a passion for these domains from a young age, students will have the opportunity to explore the diversity and richness of global cultures, leading to a more well-rounded and culturally aware society.

Another driving force behind the project is to cater to the needs of students and individuals who are seeking a comprehensive resource for learning global geography, history, and related topics. Existing students in these disciplines can benefit from the platform by accessing a wealth of information, interactive quizzes, and engaging content that complements their academic studies. Additionally, job seekers preparing for interviews that require a broad understanding of global topics can utilize MappingTheWorld to enhance their knowledge and improve their chances of success.

The project also recognizes the importance of fostering collaboration and knowledge sharing among users. By incorporating features such as a real-time chatting system and a question-answer forum, MappingTheWorld aims to create a vibrant community where individuals from different backgrounds can connect, discuss, and exchange ideas. This aspect of the project not only enhances the learning experience but also promotes cultural exchange, empathy, and a global perspective.

In conclusion, the motivation behind MappingTheWorld lies in the desire to revolutionize the way individuals learn about global geography, history, and humanities. By creating an interactive and engaging learning platform, the project aims to overcome the challenges of disengagement and stereotypes. Through its innovative features and comprehensive content, MappingTheWorld strives to provide a beneficial educational resource that caters to the needs of students, job seekers, and knowledge enthusiasts, ultimately contributing to a more informed, culturally aware, and interconnected society.

1.2 Overview of the Project

After opening the application the user will be able to see the following tabs:

1. **Countries:** By clicking this tab, the user will find various options such as -

a. General facts: capital, land area, population.
b. History
c. Climate
d. Culture

By clicking any of the options, the user will see a world map and by clicking any country the user will be able to see facts, history or any other information depending upon the option chosen.

2. Ocean and Seas: By choosing this option, the user will be able to click on any of the ocean or sea of the world thus viewing all the necessary information about that particular choice.

3. Sports: By clicking on the sports tab, the user will be able to choose between popular sports such as football and cricket. Upon clicking an option, the user will be able to see the football and cricket playing nations marked on the map.

4. Forms of Government: By clicking on the tab, users will be able to see the countries that follow communism or have an existing monarchy.

5. **Travel:** By clicking this tab, the countries with most iconic tourist destinations will be marked.

6. **Unions:** By clicking on this tab, the user will get some options like - SAARC, ASEAN, GCC etc and see those countries marked in the map.

7. **Quiz:** For using this feature, the user will have to sign up and login. The user can play two types of quizzes.

a. Multiple choice question

b. Gaming quiz: Two options will be of the country, oceans and seas. The user will see an unmarked map and a country or ocean name will be shown on the screen. The user has to click on the location of the country or ocean based on the name shown on the screen.

The quiz will be played in rounds and by completing each round there will be a level up and XP gained. A leaderboard of the top scorer of the month will be shown on a dashboard.

8. **Chat:** For using this feature, the user will have to sign up and login and then will see the map and number of users active from a country will be shown in the map and by sending chat requests, different people from the world will be able to communicate with each other.

CHAPTER 2

BACKGROUND OF MAPPING THE WORLD

Our project is a web application that is built on several web technologies. So, we had to do a lot of background research to have a better understanding of the full process. Let's discuss them-

2.1 Three-tier Architecture

Our project 'Mapping The World' was built on a three-tier architecture. The architecture organizes applications into three logical and physical computing tiers-

- The **Presentation** tier, or user interface;
- The **Application** tier, where data is processed; and
- The **Data** tier, where the data associated with the application is stored and managed.

2.1.1 Presentation Tier : React JS Front-end Framework

For our presentation tier, or the frontend, we are using **ReactJS framework**. It is a front-end framework which is highly efficient in keeping clean project structure and maintaining component-based modeling. It's also easier to learn than other frameworks like Angular or Vue.

To build a more interactive user interface, React provides a high-quality UI concept. It also has technical terms such as, React hooks, reducers etc. to write custom components. React comes with JSX, an optional syntax extension, which makes it possible to write your own components.

React web framework, on the other hand, is currently being utilized by famous companies including Netflix, Paypal, NASA, BBC, Lyft, and New York Times to name just a few. Thus, we used it.

2.1.2 Application Tier : FastAPI Back-end Framework

For our application tier or the backend, we are using **FastAPI Framework**. FastAPI is a high-level Python web framework. It uses REST API to receive requests from the presentation tier and return responses accordingly. It can work with any client-side framework, and can deliver content in almost any format. It has a component-based architecture model to provide higher security and scalability. In a website, a web application waits for HTTP requests from the client. When a request is received the application works out what is needed based on the URL and possibly information in POST data or GET data. Depending on what is required it may then read or write information from a database or perform other tasks required to satisfy the request.

2.1.3 Data Tier : SQLITE Database

Our database contains structured data and uses **SQLite** which is a Relational Database Management System. SQLite has been used because it provides comprehensive support for every application development need. A Unique storage-engine architecture allows database professionals to configure the SQLite database server specifically for particular applications, with the end result being amazing performance results. Thus, we used SQLite.

2.2 REST API

The FastAPI Framework uses REST API to receive requests from the presentation tier and return responses accordingly. The REST API is an Application Programming Interface that allows interaction of RESTful web services. The full form of REST API is Representational State Transfer Application Programming Interface. It completes CRUD operations through POST, GET, PUT, DELETE etc. methods. It helps connect the front-end with the back-end and thus store data securely.

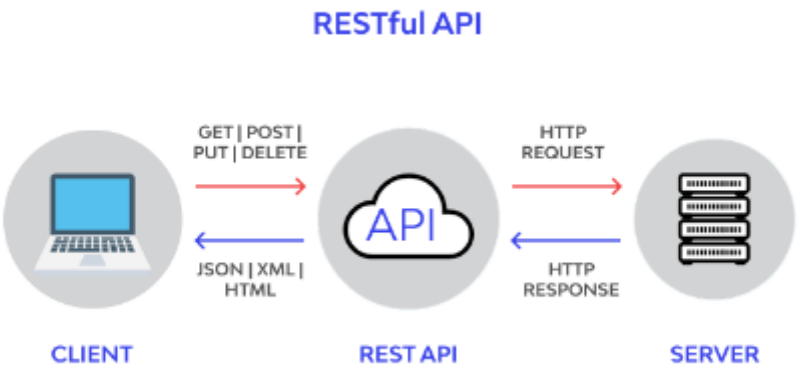

FIGURE-1 : RESTful API Mechanism

CHAPTER 3

DESCRIPTION OF Mapping The World

Mapping The World is a website with many features to learn all about the countries of the world including a real time chatting system, quiz, question answer forum for creating an interactive platform for anyone around the world.

The website consists of five major modules, which are :

1. Sign Up/ Login
2. Countries
3. Ocean & Seas
4. Union
5. Travel
6. Sports
7. Climate
8. Download
9. Quiz
10. Game
11. Chat

After opening the application the user will be able to see the following tabs:

3.1 Sign Up/ Login:

There will be two types of users for this application.

a. **Admin:** Predefined account for admin will be created and admin will be able to change any information within the application and manage user profile.

b. **User:** Firstly, the user needs to register with necessary information (Full Name, User Name, Email id, Country, Password). After registration, the user can login by providing the correct username and password.

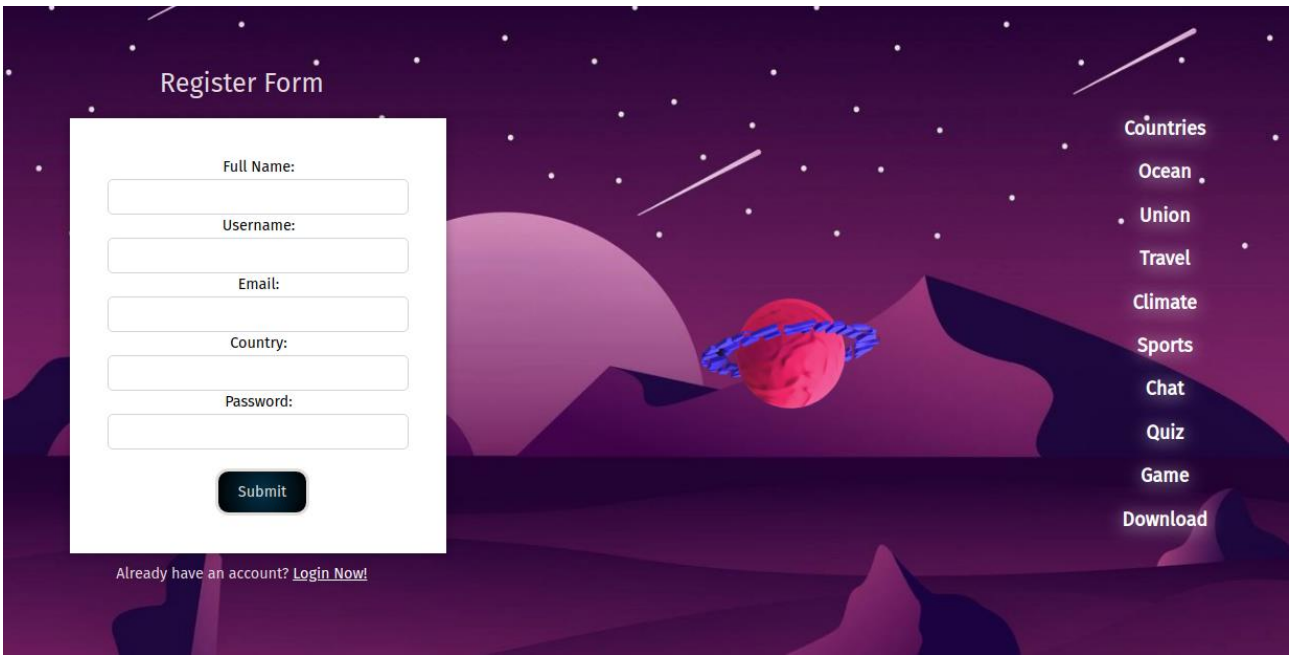

FIGURE-2 : Sign Up Page

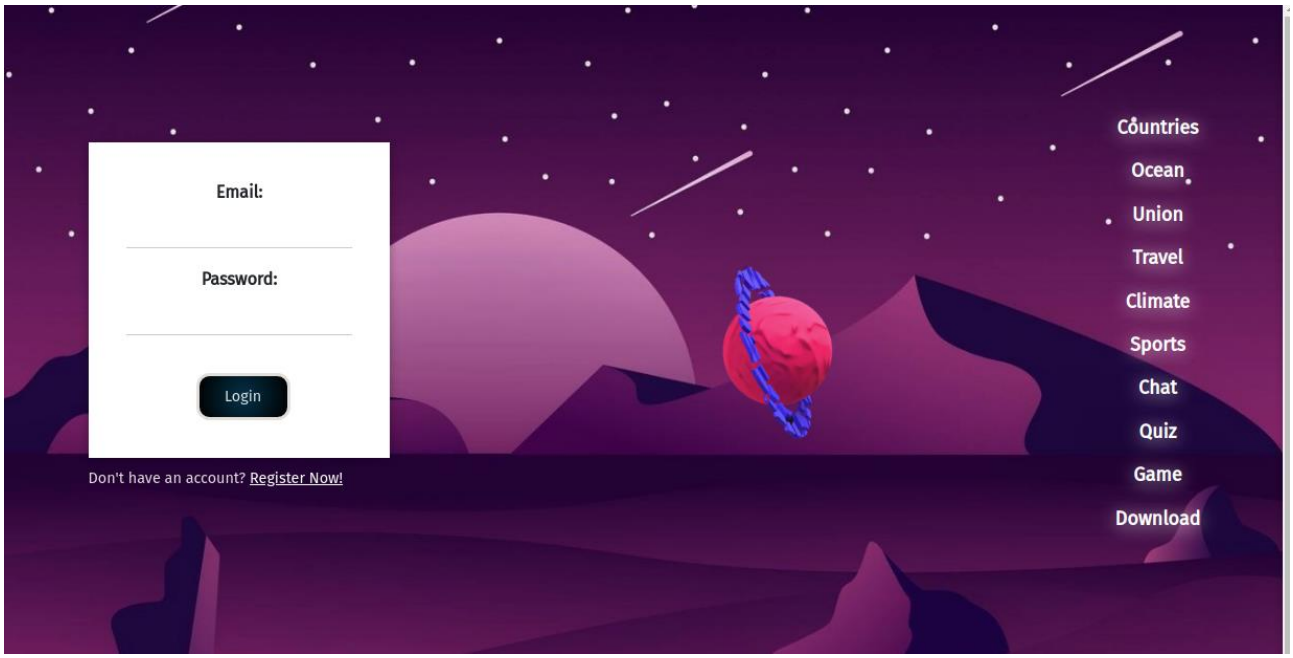

FIGURE-3 : Login Page

3.2 Countries:

By clicking this tab, the user will find various options such as -

- **c.** Government
- **c.** History
- **c.** Geography

By clicking any of the options, the user will see a world map and by clicking any country the user will be able to see facts, history or any other information depending upon the option chosen.

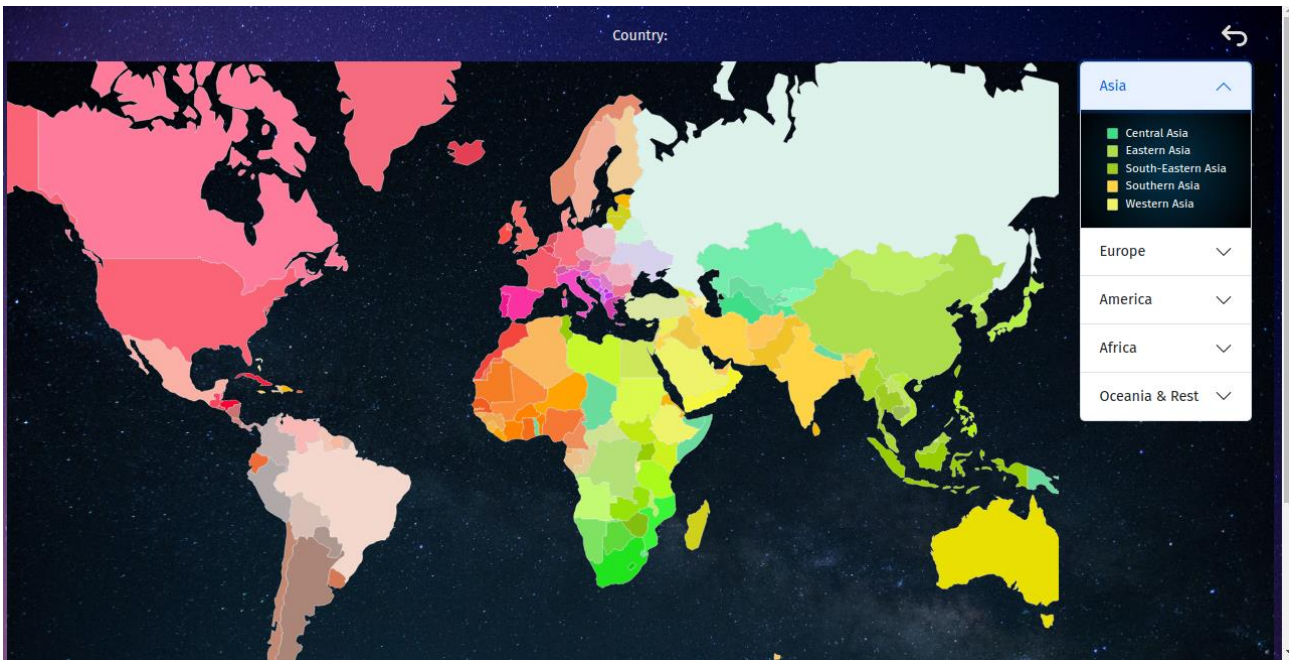

FIGURE-4 : Countries

3.3 Ocean:

By choosing this option, the user will be able to click on any of the ocean or sea of the world thus viewing all the necessary information about that particular choice.

FIGURE-5 : Ocean

3.4 Union:

By clicking on this tab, the user will get some options like - SAARC, ASEAN, GCC, Nordic countries, Baltic countries etc and see those countries marked in the map.

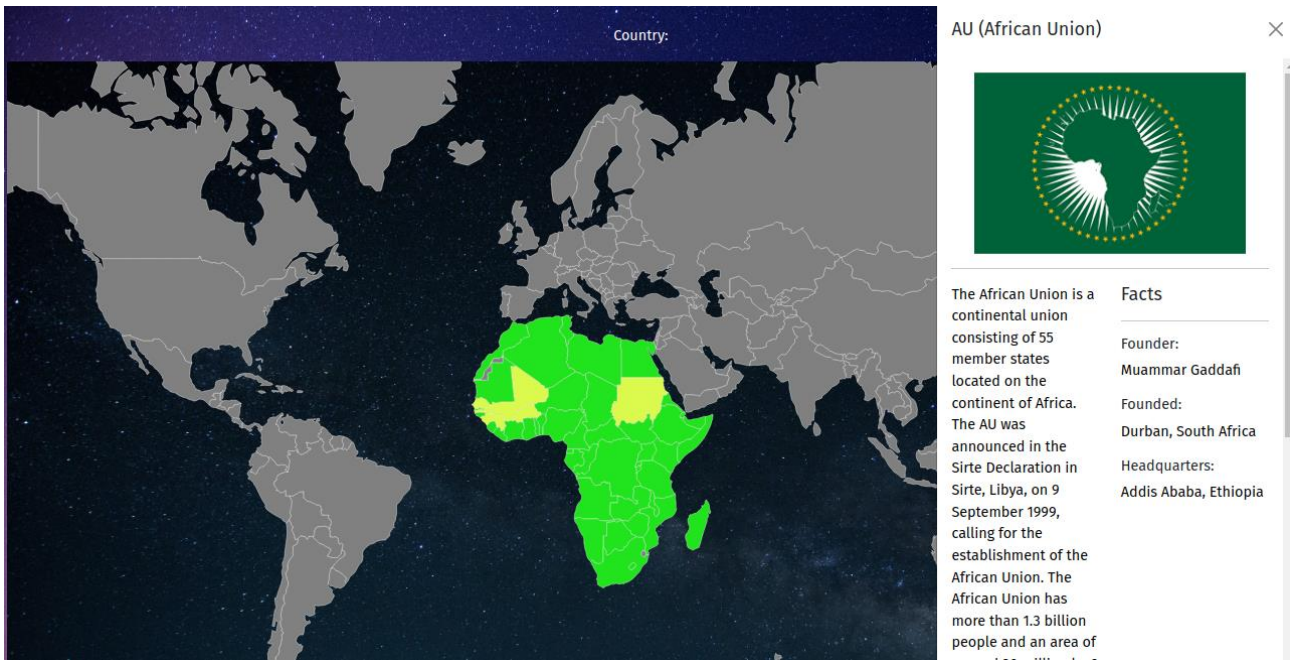

FIGURE-5 : Union

3.5 Travel:

By clicking this tab, the countries with most iconic tourist destinations will be marked.

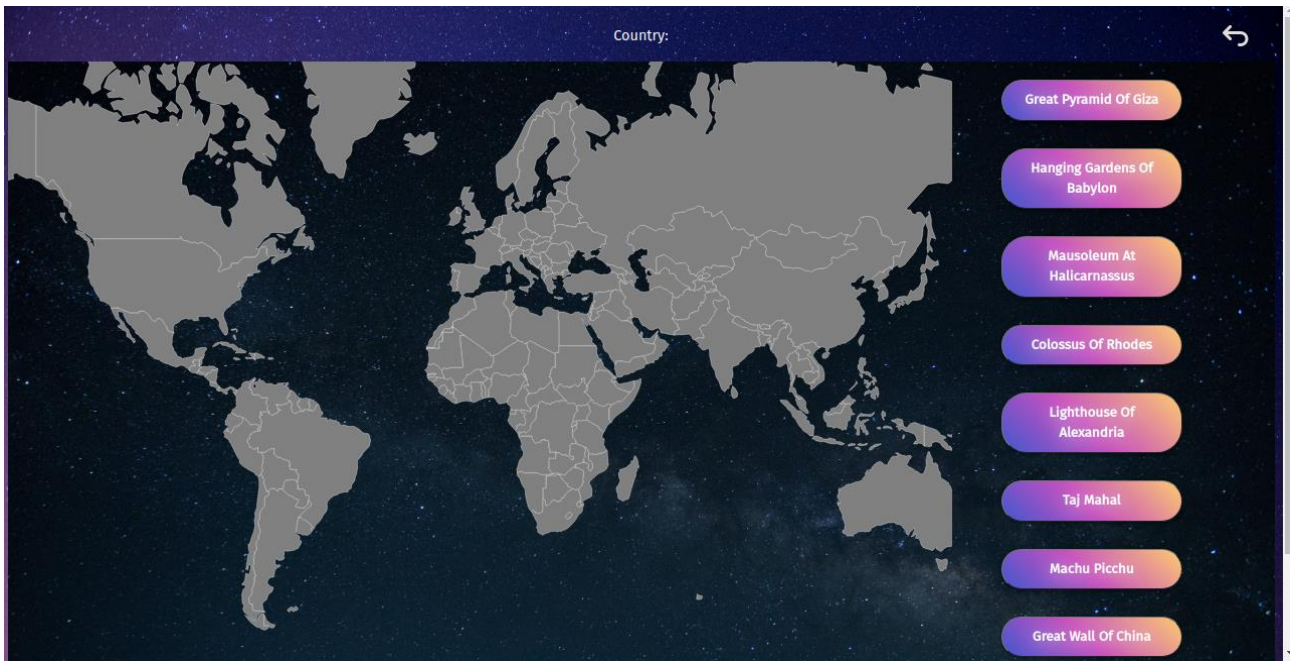

FIGURE-6 : Travel

3.6 Sports:

By clicking on the sports tab, the user will be able to choose between popular sports such as football and cricket. Upon clicking an option, the user will be able to see the football and cricket playing nations marked on the map.

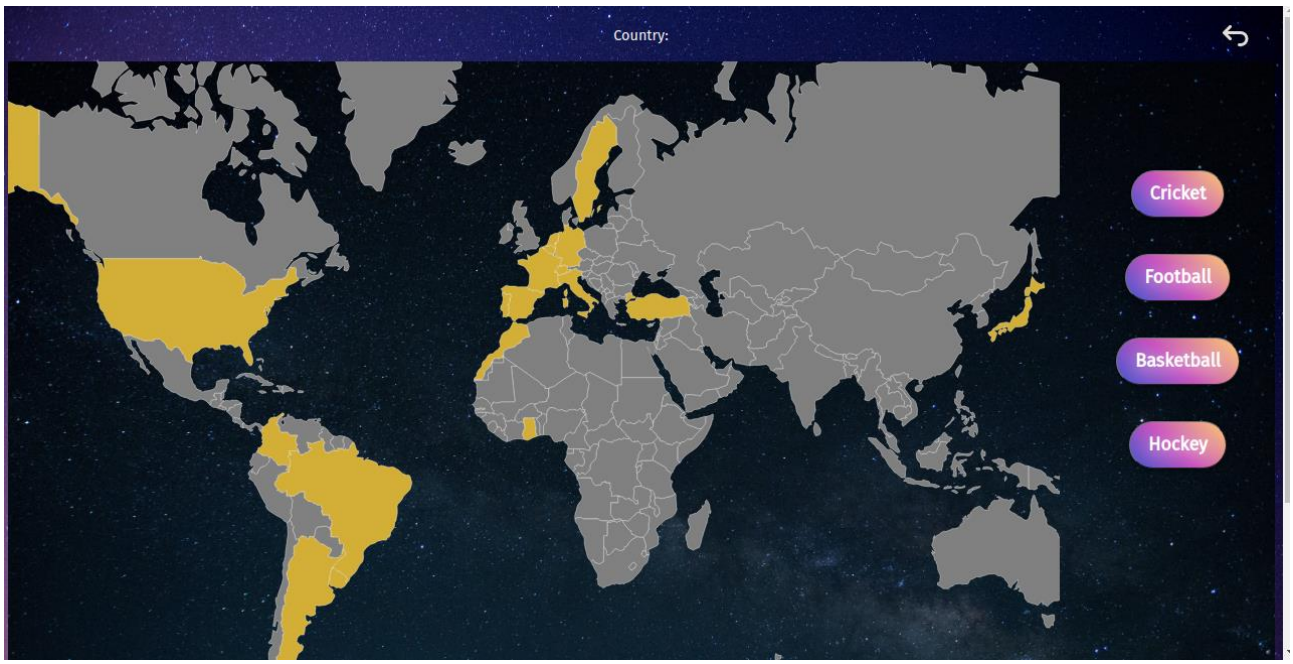

FIGURE-7 : Sports

3.7 Climate:

By clicking this tab, users can see the climate of the country.

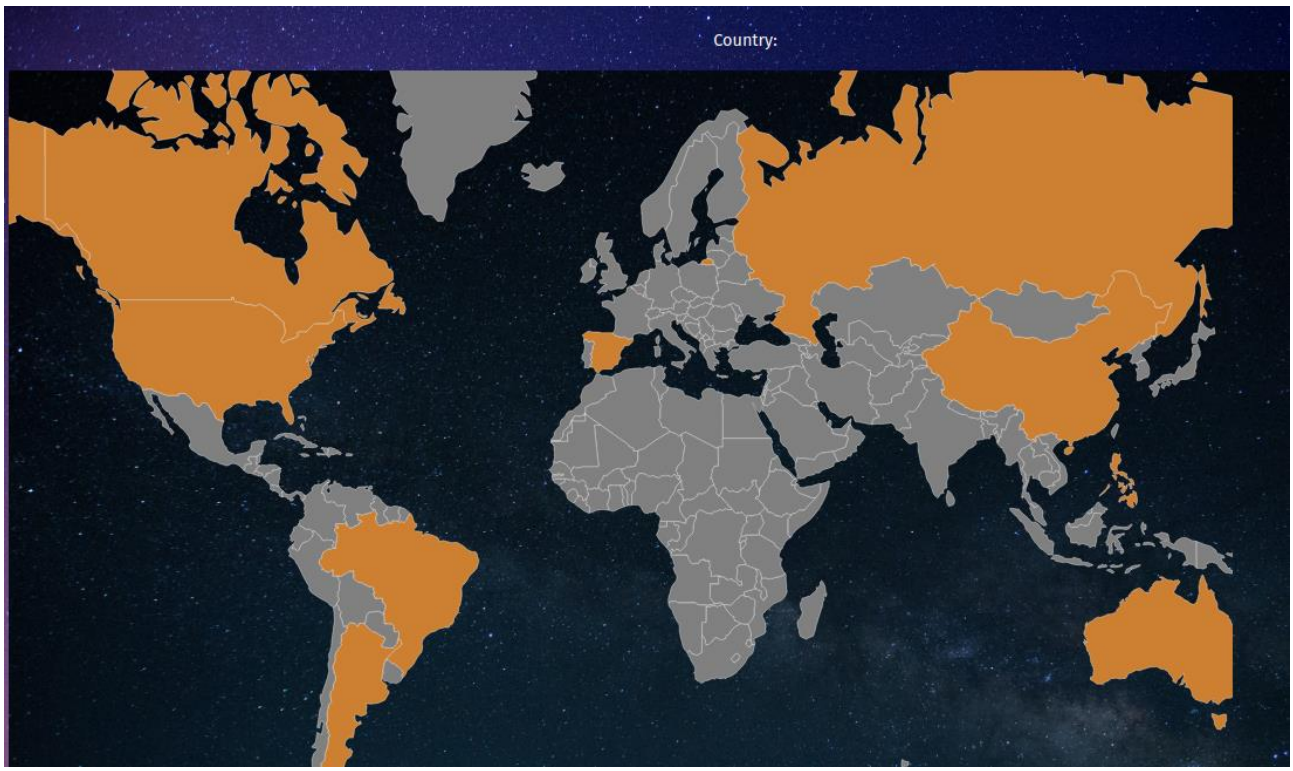

FIGURE-8 : Climate

3.8 Download

Users can download capital, currency, language, religion.

FIGURE-9 : Download

3.9 Quiz:

For using this feature, the user will have to sign up and login. Admin will set some predefined questions with 4 options. User needs to guess the correct answer. After guessing 80% right within time, user level will be updated.

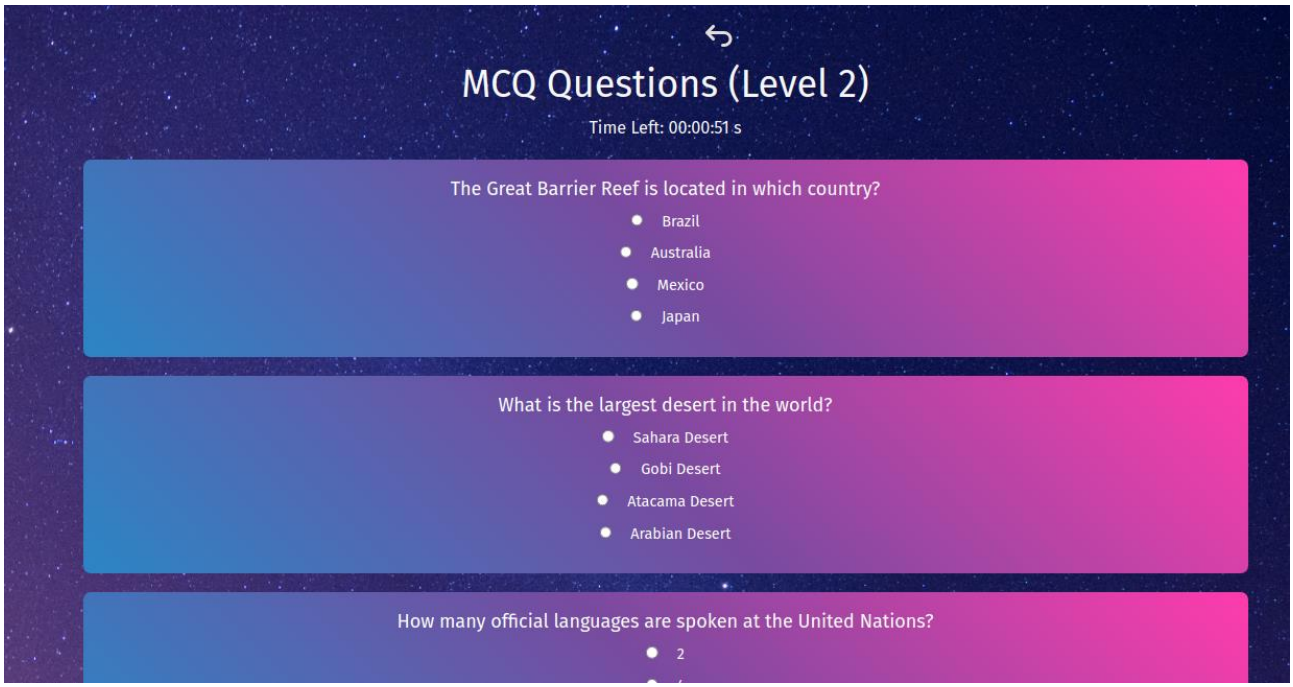

FIGURE-10 : Quiz

3.10 Game:

The user will see an unmarked map and a country name will be shown on the screen. The user has to click on the location of the country based on the name shown on the screen.

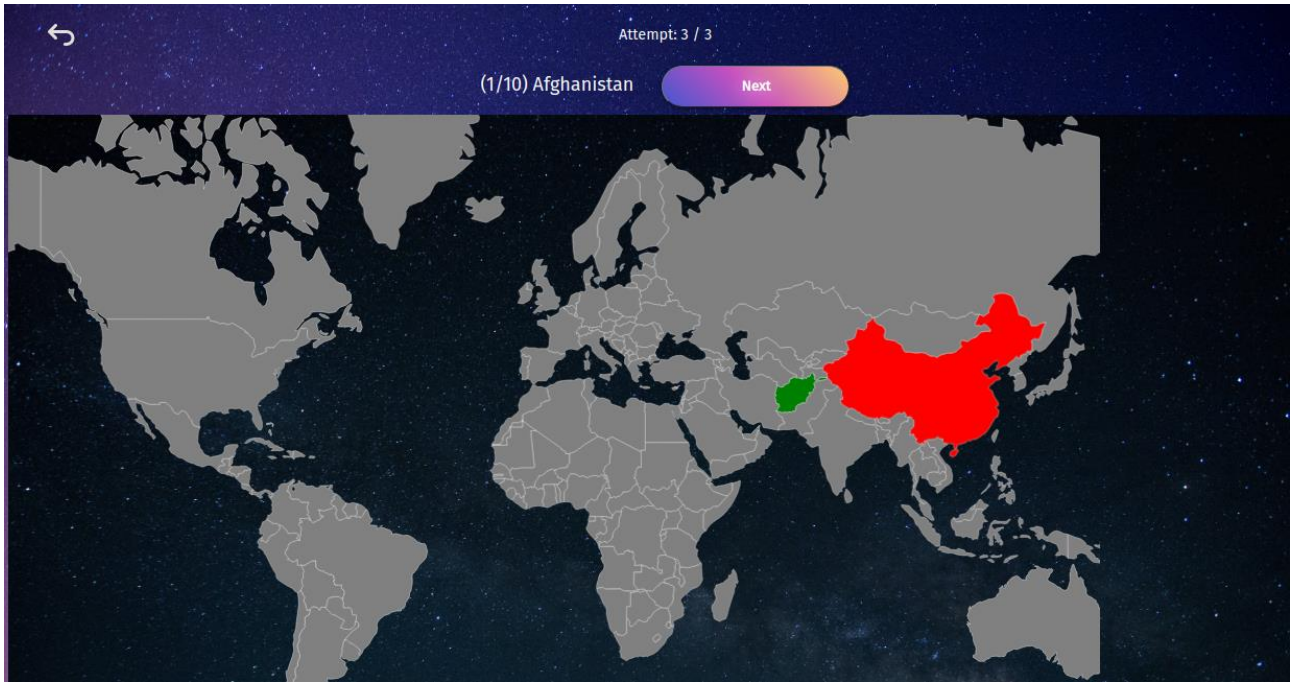

FIGURE-11 : Game

3.11 Chat:

For using this feature, the user will have to sign up and login and by sending chat requests, different people from the world will be able to communicate with each other.

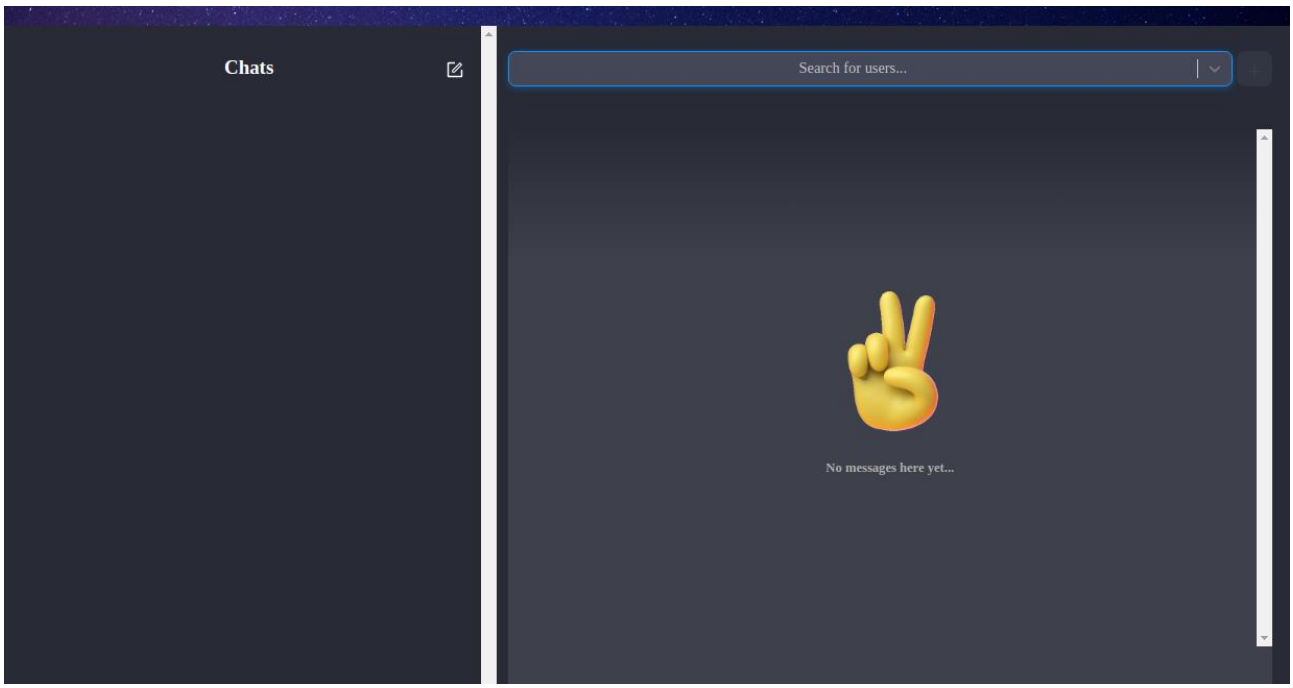

FIGURE-12 : Chat

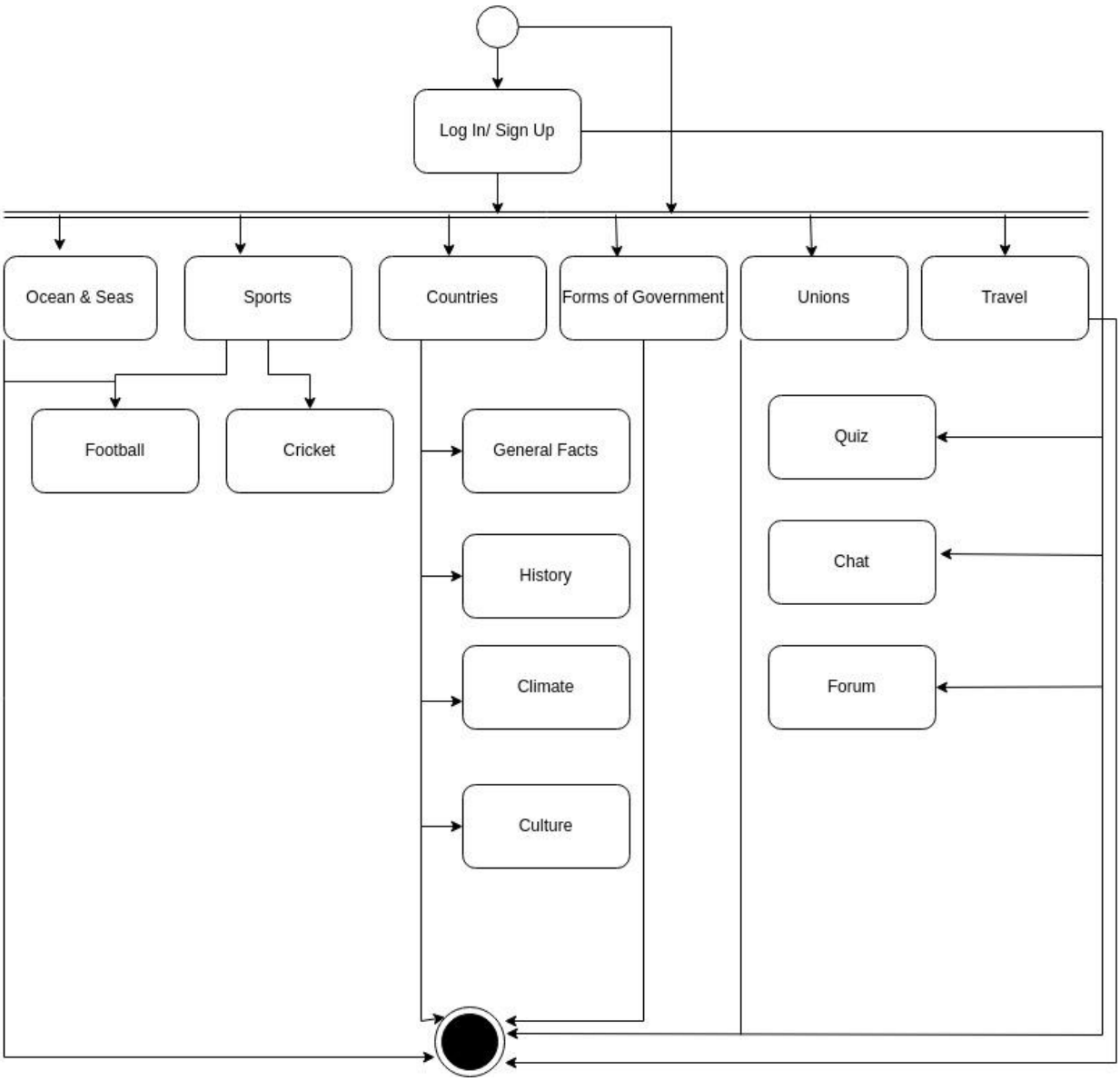

Fig13: Mapping The World

CHAPTER 4

IMPLEMENTATION & TESTING

4.1 Front-End Implementation

We've used a JavaScript based framework, React JS for front-end development. React is a very popular framework and has component-based modeling to implement user interface with more ease.

4.1.1 Styling

For styling, we've incorporated bootstrap 5 library with the application. We've also used different libraries of react to make our application look more attractive. For example-

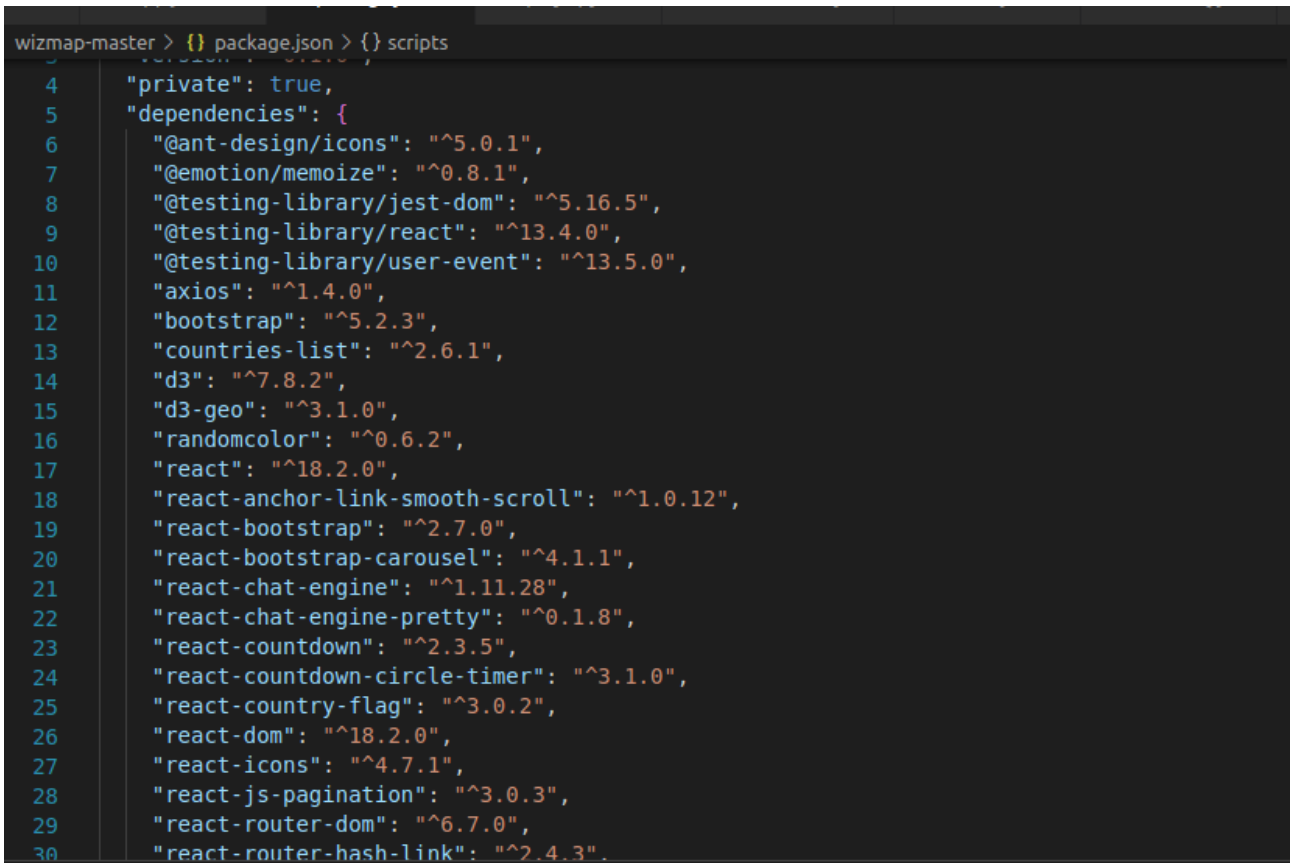

FIGURE-14 : Dependencies from package.json

We used react-icons library to fetch icons from different icons providing services such as, font-awesome, material io etc. We also used react-calendar and react-datepicker to show a more organized version of schedule calendars. Other than date, react-toastify has been used to show more decorated alert messages. We used different components of reactstrap as well for easy readiness of the user interface. We've added fonts from google fonts library.

4.1.2 Components Structure

Our application has following components implementing various features-

Component Name	Functions
Auth	It contains authentication related works.
Chat	It contains chat features.
Climate	It contains climate related info.
DownloadInfo	It contains various information regarding the capital, currency etc.
Game	It contains game features.
Leaderboard	It contains two types of leaderboard. One is for a quiz and another one is for a game.
MapView	It contains map related info.
Menu	It shows the menu.
Modal	It contains modal.
Navbar	It includes a navbar.
Ocean	It contains Ocean feature.
Quiz	It includes a quiz with a gamification feature.
Space	It has space feature.

Travel	It includes travel related info.
Union	It has the union feature.
WorldMap	It has the react simple maps.

TABLE-1 : Component Functionalities

4.1.3 Testing

Front-end development is somewhat purely related with the user experience. As much as the user suffers due to server side issues, the client side is highly connected with the user satisfaction. So, it's important for the users to have smooth exchange of the interfaces. Thus, software testing is necessary.

We did unit testing in some components of our application to solve bugs several times. There were some bug fixes related to routing, generating calculative results etc. It appears that our recommendation system needed more valid data from actual users to be able to provide more efficient results. So, we tested our application with a bunch of dummy datasets to prove its credibility.

4.2 Back-End Implementation

We've used FastAPI as our back-end framework, which is a python language based framework.

4.2.1 Sign Up & Login

The backend of the signup and login system in fastapi was implemented using django's built-in jwt token. This was used to create a token-based login system. We implemented a FastAPI User Model in our core app which adds the following fields, fullname, username, email, country, password. When a POST request is sent to the url http://localhost:8000/user/createUser" by filling out the user information a new user is added to the database with the given information.

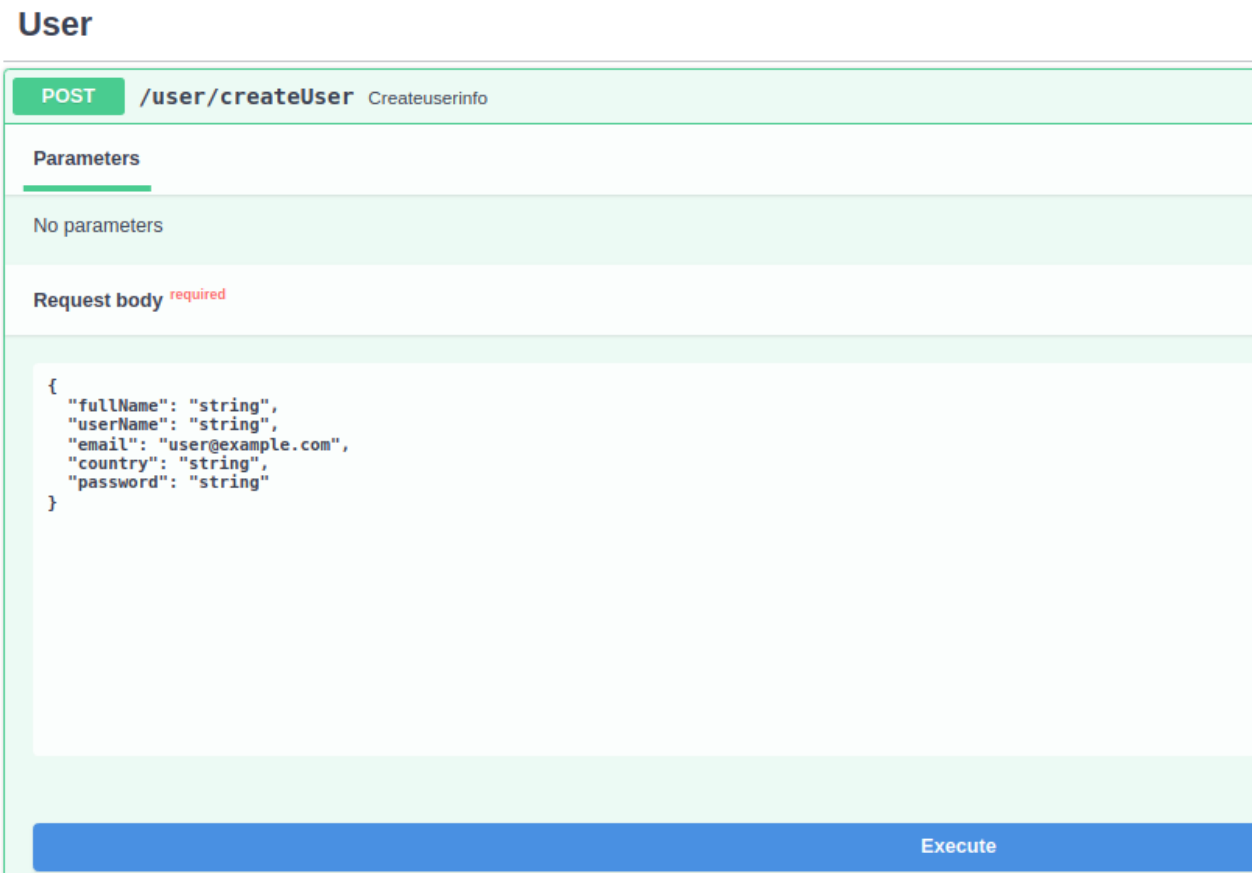

FIGURE-15 : Sign Up Completion Using Swagger

While logging in, a post request is sent to the url http://localhost:8000/login and a refresh and access token is returned as a response. The access token can be used as a request header to gain permissions to access the data related to the user from the backend.

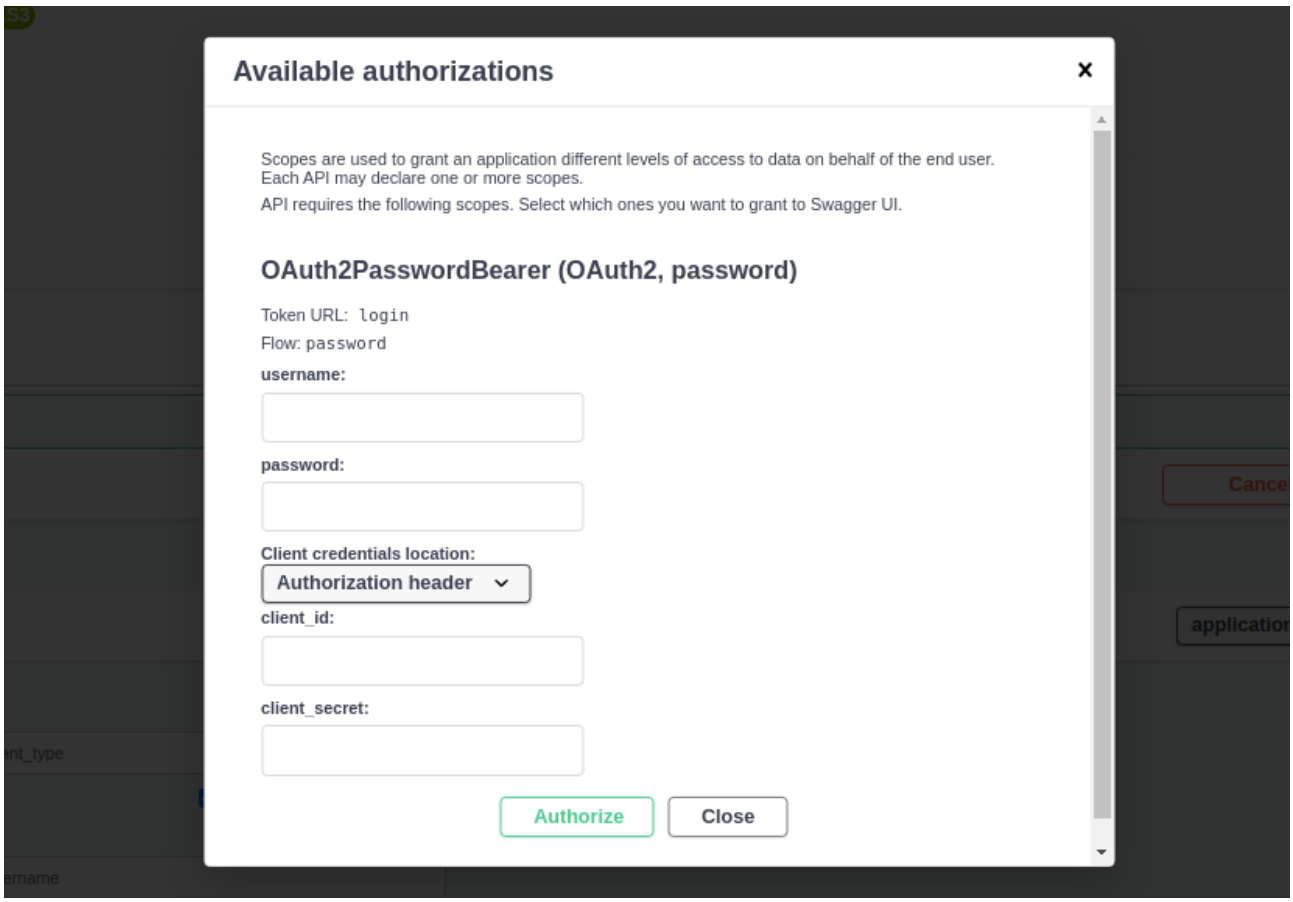

FIGURE-15 : Authorization

4.2.2 MCQ

MCQ is mostly handled by the backend of our software. Firstly, a POST request is sent to the url http://127.0.0.1:8000/mcq/createMCQ .

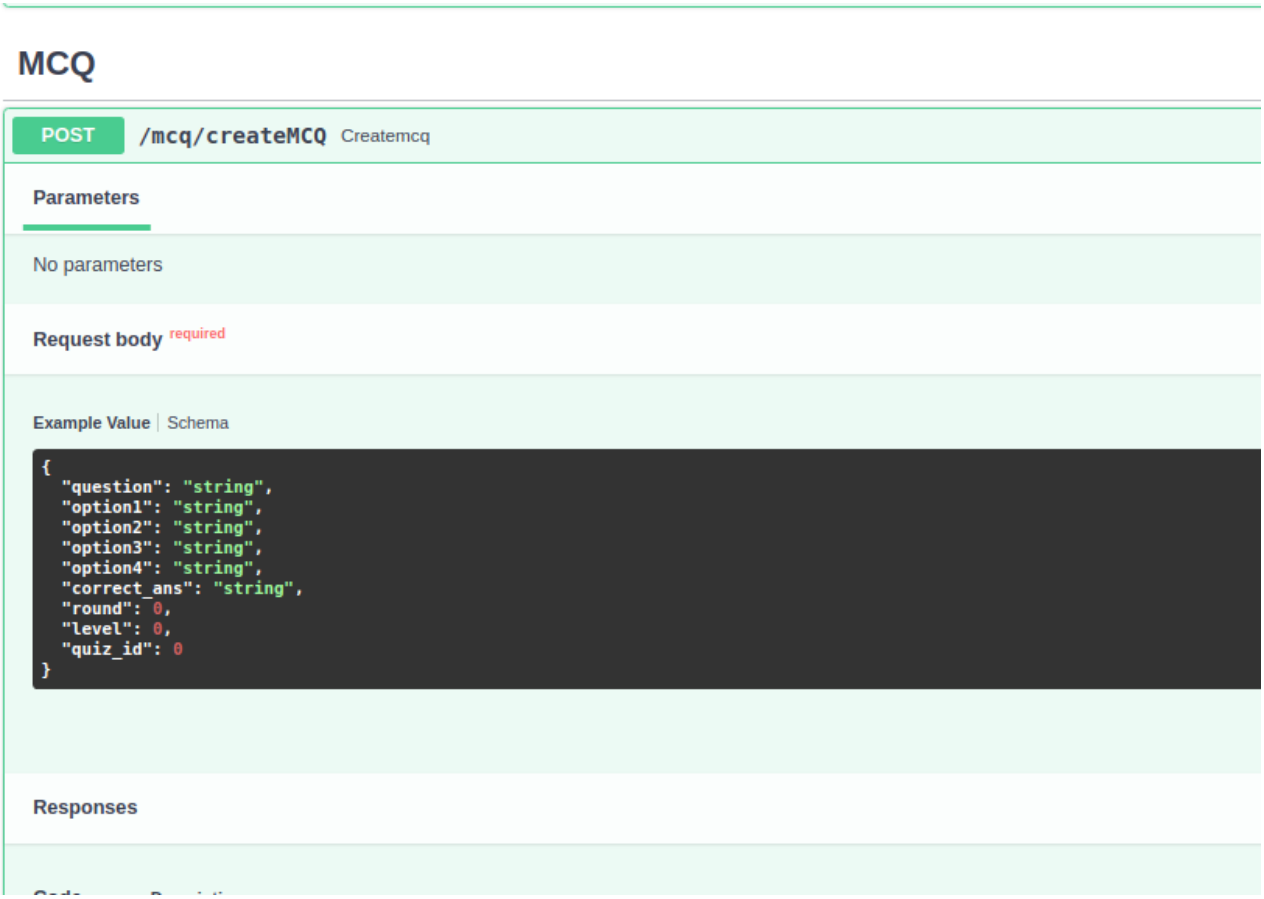

FIGURE-16 : Create MCQ

```python
#Create mcq questions
@router.post('/createMCQ', response_model=schema.mcq)
def createMCQ(request: schema.mcq, db: Session=Depends(database.get_db)):
    new_question = models.mcq(question=request.question, option1=request.option1,
                              option2=request.option2, option3=request.option3,
                              option4=request.option4, correct_ans=request.correct_ans,
                              round=request.round, level=request.level, quiz_id=request.quiz_i


    db.add(new_question)
    db.commit()
    db.refresh(new_question)
    return new_question
```

FIGURE-17 : Create MCQ API

```python
#Get question based on level
@router.get('/{level}/getSpecificMCQ', response_model=List[schema.mcq])
def showSpecificQuestions(level:int, db: Session=Depends(database.get_db)):
    question = db.query(models.mcq).filter(models.mcq.level == level).all()
    print(question)
    if not question:
        raise HTTPException(status_code=status.HTTP_404_NOT_FOUND,
                            detail=f"Question for level {level} is not available")

#Show all question
@router.get('/allMCQ', response_model=List[schema.showMcq])
def allQuestions(db: Session = Depends(database.get_db)):
    all_mcq = db.query(models.mcq).all()
    return all_mcq
```

FIGURE-18 : Showing questions API

```python
#Update questions
@router.put('/{id}/updateMCQ', status_code=status.HTTP_202_ACCEPTED)
def updateMcq(id:int, request: schema.updateMcq, db:Session = Depends(database.get_db)):
    question = db.query(models.mcq).filter(models.mcq.id == id)
    if not question.first():
        raise HTTPException(status_code=status.HTTP_404_NOT_FOUND,
                            detail=f"Question with id {id} not found")

    question.update(request.dict(exclude_unset=True))
    db.commit()
    return 'updated'

#Delete question
@router.delete('/{id}/deleteMCQ')
def destroy(id:int, db:Session=Depends(database.get_db), current_user: schema.userInfo=Depe
    question = db.query(models.mcq).filter(models.mcq.id==id)
    if not question.first():
        raise HTTPException(status_code=status.HTTP_404_NOT_FOUND,
                            detail=f"Question with id {id} not found")
    question.delete(synchronize_session=False)
    db.commit()
    return 'done'
```

FIGURE-19 : Update & Delete MCQ API

A POST route at "/createMCQ" is defined with the endpoint function createMCQ. The function expects a request body (request) of type schema.mcq (a Pydantic schema).

A new models.mcq object is created based on the request data. The object is added to the database session (db), committed, and refreshed to obtain the updated object. The new question is returned as the response.

A GET route at "/{level}/getSpecificMCQ" that expects a path parameter level (an integer). It queries the database for MCQ questions with the specified level and returns them as a list. If no questions are found, it raises an HTTP 404 error.

A GET route at "/allMCQ" that queries the database for all MCQ questions and returns them as a list.

A PUT route at "/{id}/updateMCQ" that expects a path parameter id (an integer) to identify the question to update. It also expects a request body of type schema.updateMcq. It queries the database for the question with the specified ID, updates it with the provided data, and commits the changes. If no question is found with the given ID, it raises an HTTP 404 error. It returns a string "updated" as the response.

A DELETE route at "/{id}/deleteMCQ" that expects a path parameter id (an integer) to identify the question to delete. It also expects a current_user dependency obtained from the oauth2.get_current_user function. It queries the database for the question with the specified ID, deletes it, commits the changes, and returns "done".

CHAPTER 5

USER MANUAL of Mapping The World

Our user interface will have a homepage to start with like every other web application. The homepage will have the following-

- Navigation Bar
- Cover Page
- Exploration Section

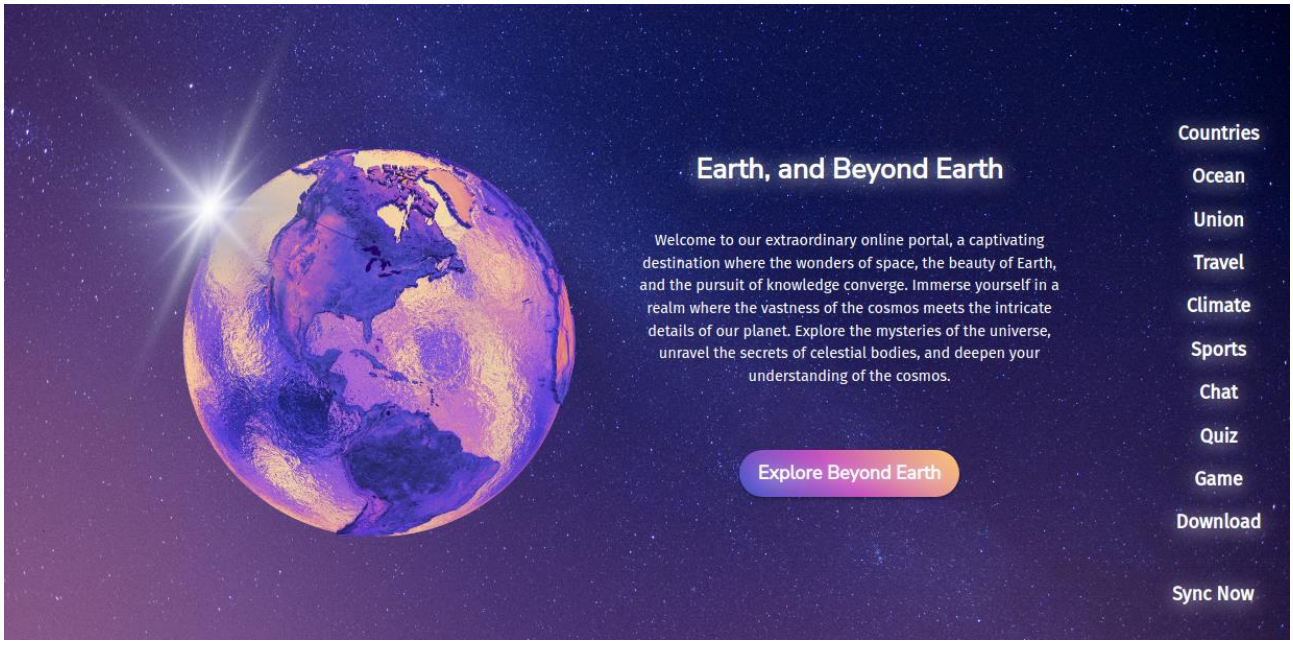

FIGURE-20 : Homepage of Mapping The World

5.1 Sign Up/ Login:

There will be two types of users for this application.

a. **Admin:** Predefined account for admin will be created and admin will be able to change any information within the application and manage user profile.

b. **User:** Firstly, the user needs to register with necessary information (Full Name, User Name, Email id, Country, Password). After registration, the user can login by providing the correct username and password.

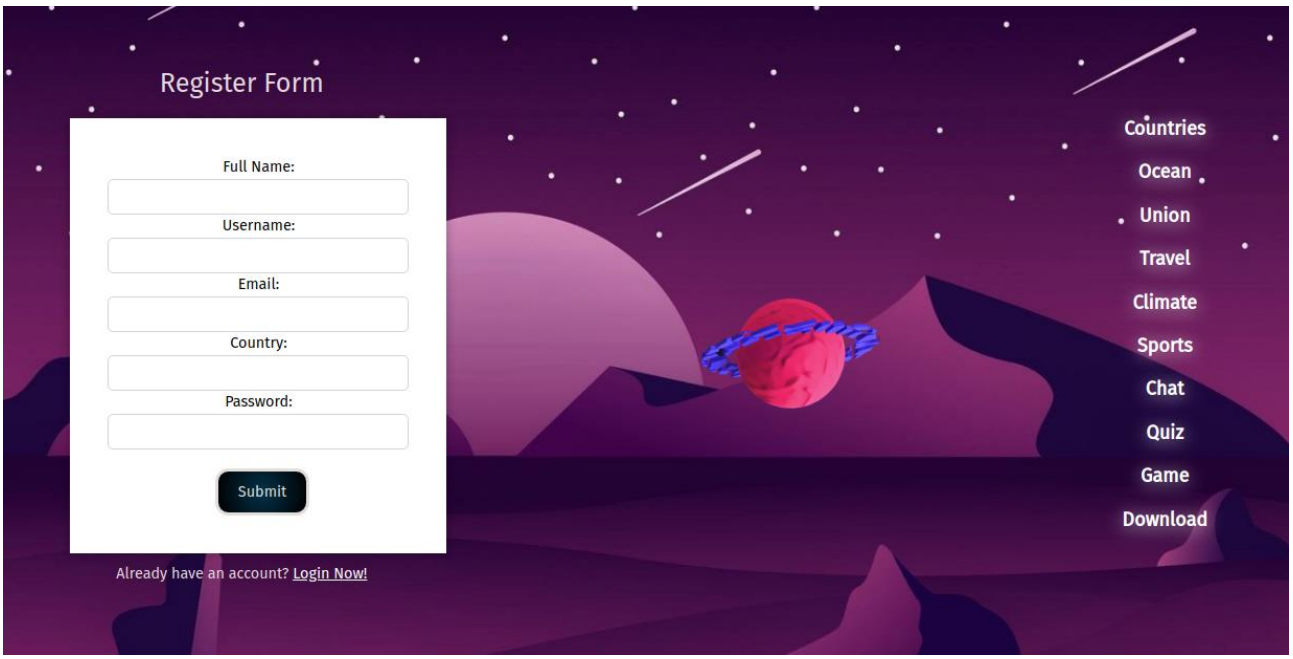

FIGURE-21 : Sign Up Page

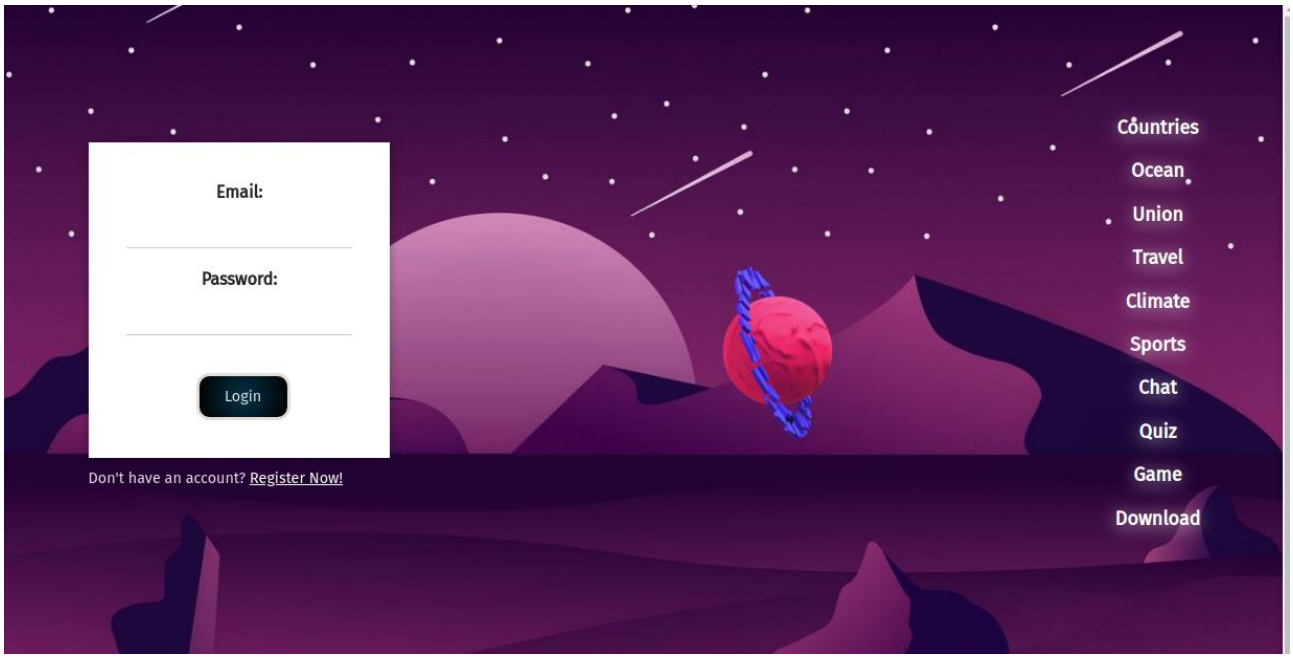

FIGURE-22 : Login Page

5.2 Countries:

By clicking this tab, the user will find various options such as -

- **a.** Government
- **b.** History
- **c.** Geography

By clicking any of the options, the user will see a world map and by clicking any country the user will be able to see facts, history or any other information depending upon the option chosen.

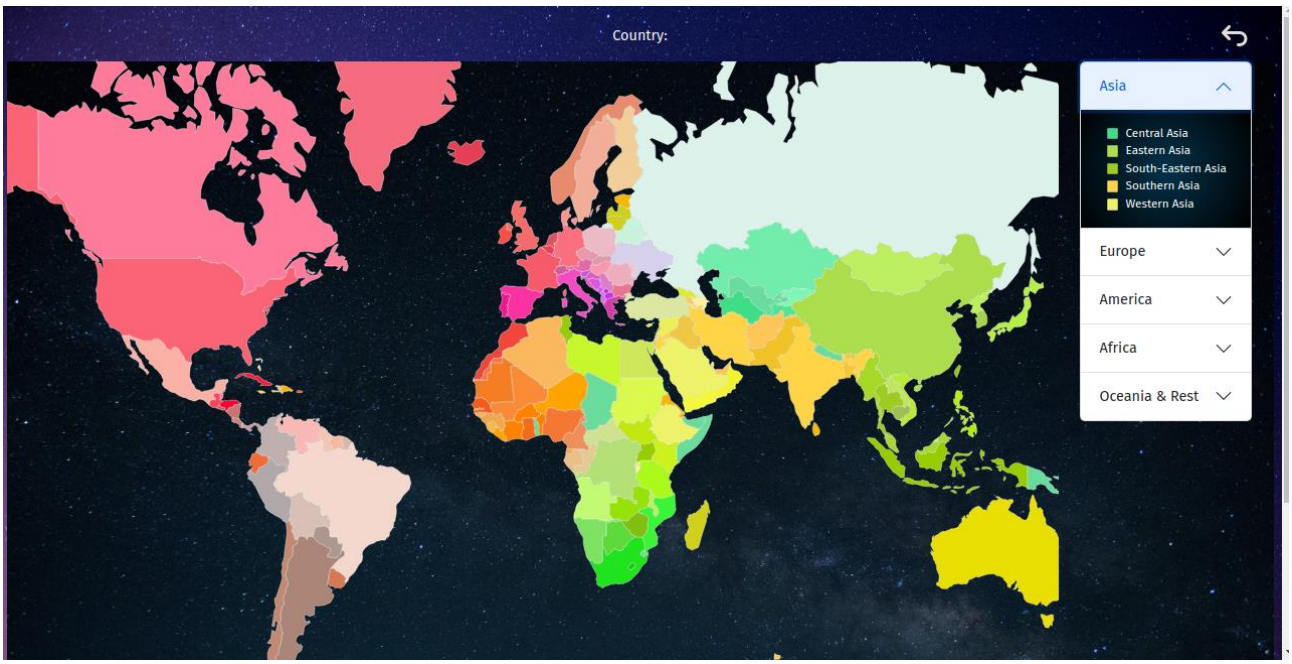

FIGURE-23 : Countries

5.3 Ocean:

By choosing this option, the user will be able to click on any of the ocean or sea of the world thus viewing all the necessary information about that particular choice.

FIGURE-24 : Ocean

5.4 Union:

By clicking on this tab, the user will get some options like - SAARC, ASEAN, GCC, Nordic countries, Baltic countries etc and see those countries marked in the map.

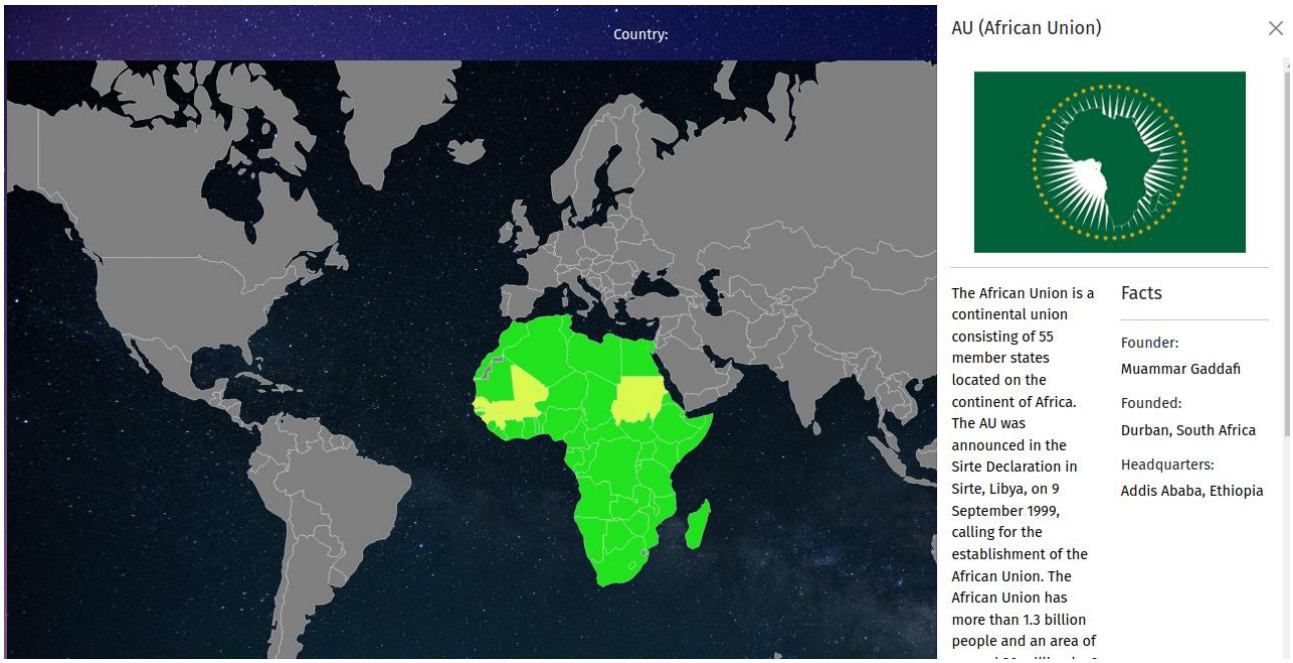

FIGURE-25 : Union

5.5 Travel:

By clicking this tab, the countries with most iconic tourist destinations will be marked.

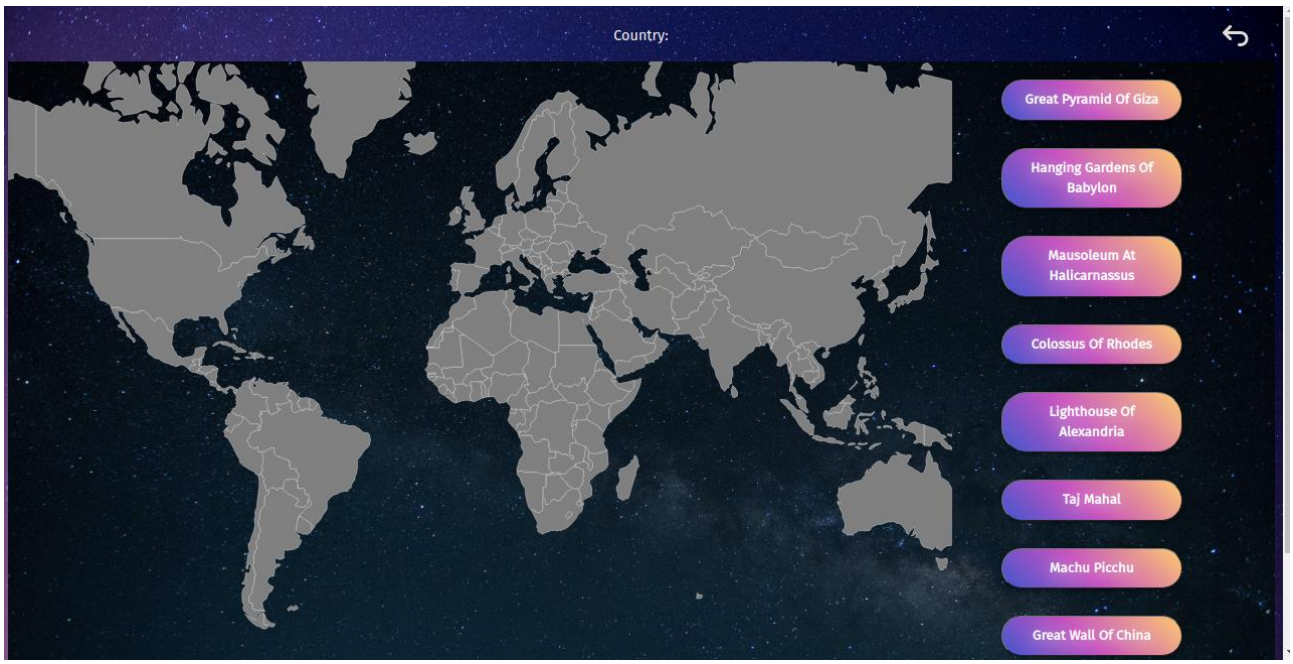

FIGURE-26 : Travel

5.6 Sports:

By clicking on the sports tab, the user will be able to choose between popular sports such as football and cricket. Upon clicking an option, the user will be able to see the football and cricket playing nations marked on the map.

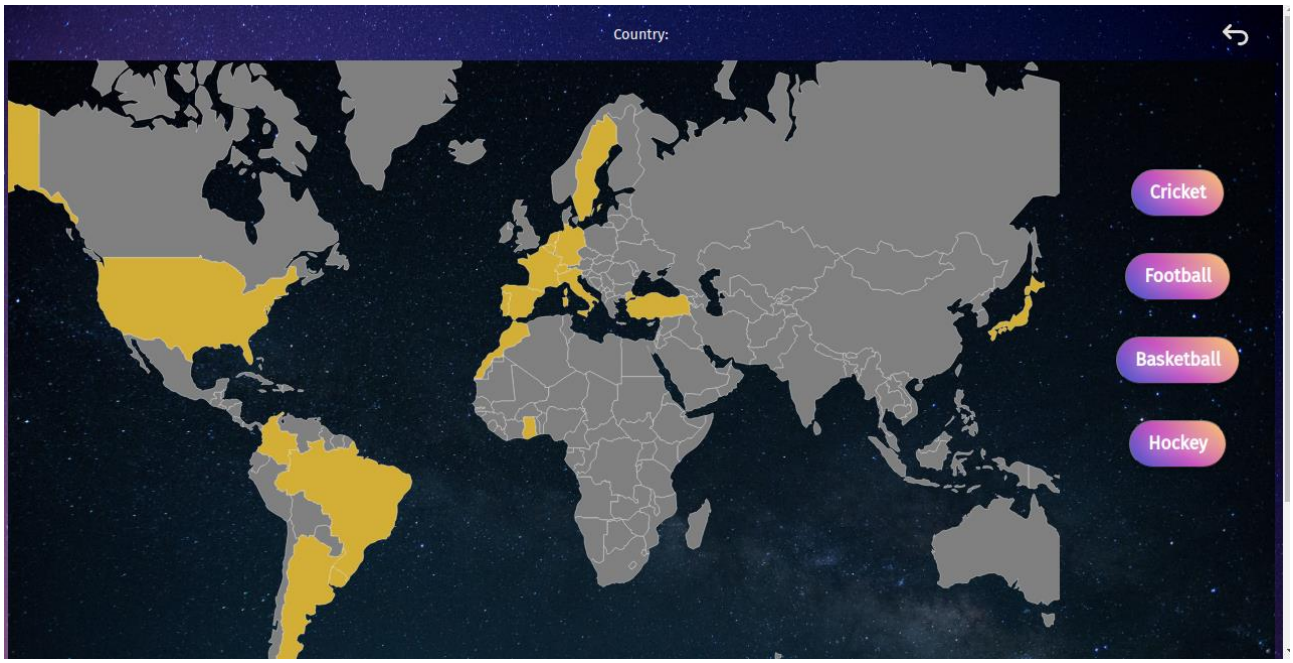

FIGURE-27 : Sports

5.7 Climate:

By clicking this tab, users can see the climate of the country.

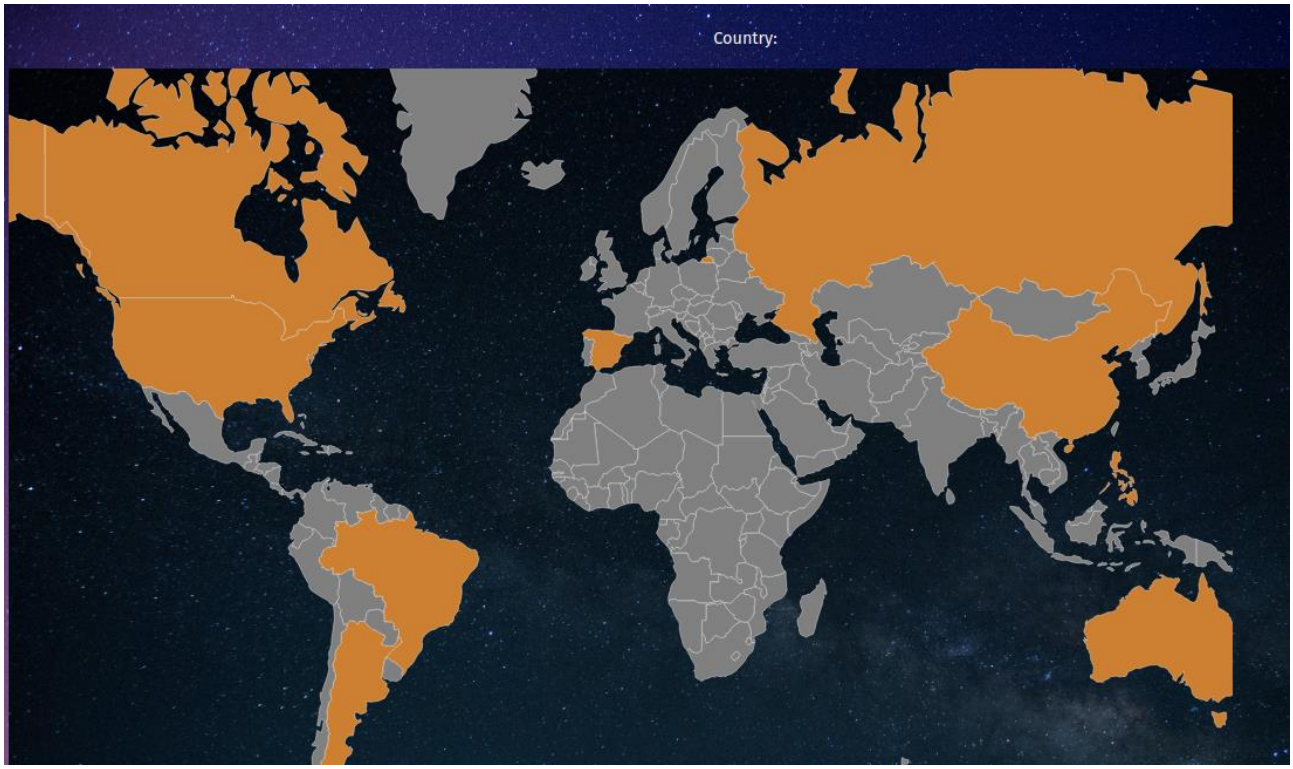

FIGURE-28 : Climate

5.8 Download:

Users can download capital, currency, language, religion.

FIGURE-29 : Download

5.9 Quiz:

For using this feature, the user will have to sign up and login. Admin will set some predefined questions with 4 options. User needs to guess the correct answer. After guessing 80% right within time, user level will be updated.

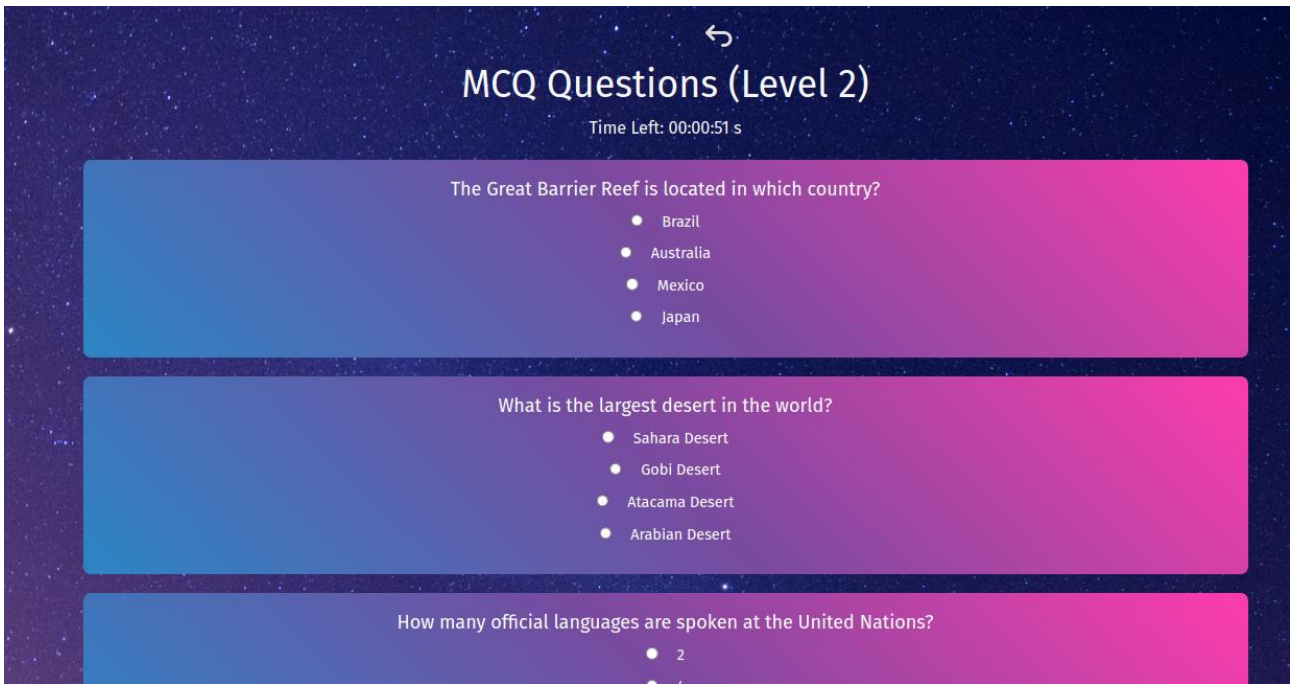

FIGURE-30 : Quiz

5.10 Game:

The user will see an unmarked map and a country name will be shown on the screen. The user has to click on the location of the country based on the name shown on the screen.

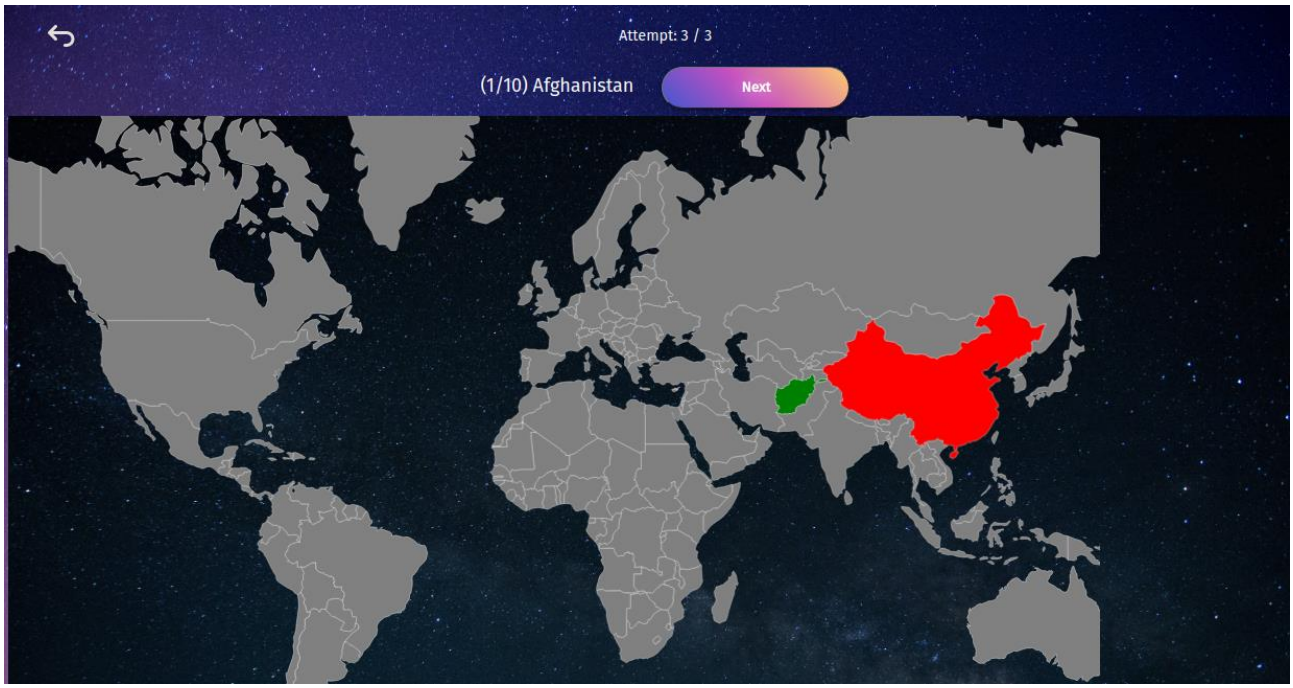

FIGURE-31 : Game

5.11 Chat:

For using this feature, the user will have to sign up and login and by sending chat requests, different people from the world will be able to communicate with each other.

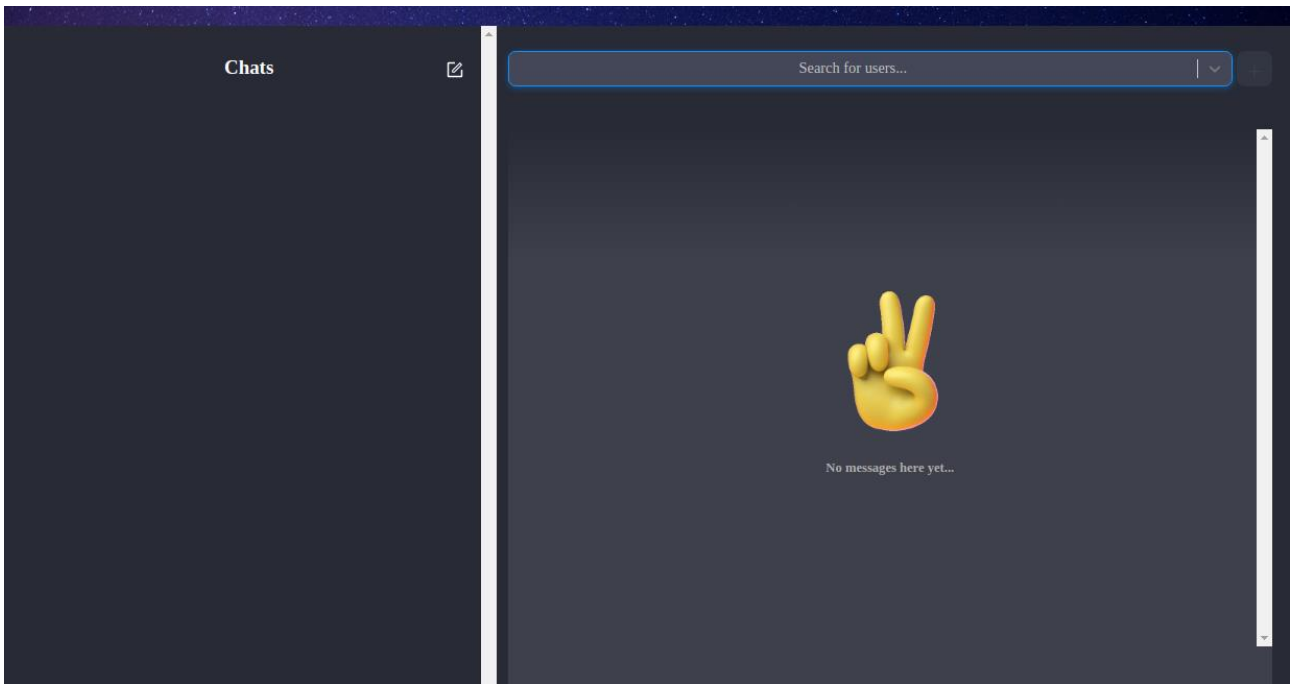

FIGURE-32 : Chat

CHAPTER 6

CHALLENGES FACED in Implementing

As software engineering students, we had prior knowledge about web technologies. Although, the concept of three-tier architecture and frameworks were something that we had to learn upon during the project timeline. Few of the challenges we faced during our three-month project were-

6.1 Learning to Collaborate & Distribute

As we've followed the three-tier architecture, it was given that we've to maintain a front-end or client side with a back-end or server side along with an additional database management system. To follow through the technologies, we had to brainstorm and derive a plan to do research, background study and finally coming up with appropriate technology that may go with our plan were all a bit hectic. But as soon as we derived our general plan, we were able to catch up with previous stallments.

6.2 Requirements Engineering & Data Collection

To build a product, finding its requirements and specifications, are probably the most primary thing to do. However, in our case, we didn't rest only by collecting requirements. Our application is heavily based on real-time data as we've a recommendation system which is built upon past actions. So, we had to go from vendor to vendor to collect real data. Again, our system had two types of users and five types of sub users. So, acting upon different scenarios was a real difficult thing to do while doing requirement analysis.

However, we managed to collect enough datasets, but also needed to rely on dummy datasets as well for testing purposes.

6.3 Version Updates

While implementing the front-end, React JS has different component structures to make things happen. One of them is- react-router-dom. In our case, what happened is, in React JS version 18, there was a significant update in the syntax. In the react-router-dom library, a syntax change happened from v5 to v6. V6 is only runnable from React JS 18. But the learning process we were following had documentation of React JS versions lower than 17. So, every time we loaded, it returned an error and a blank page because of the react-router-dom v5 version. Similar problems happened with multiple libraries in both frontend and backend. This was a drawback for us. However, we learned from mistakes and looked for the appropriate documentation from then on.

FIGURE-33 : Syntax Changes of react-router-dom ("Upgrading from v5")

6.4 CRUD Operations

We used React JS as front-end and FastAPI framework as backend. React is a JavaScript based language framework, whereas, FAstAPI has Python language as its implementation criteria. So, at the end of the day, while performing CRUD operations, we faced a bit of difficulty as GET, POST, PUT methods have different types of implementations in both languages.

To overcome the problem, we had to study a bit more and then we generated a set of functions that could easily translate the methods. Thus, we connected the front-end to the back-end.

6.5 Back-end Complications

One of the key challenges faced during the development of the project was to learn the FastAPI framework which was used to build the backend. The FastAPI framework uses models to build database structure, serializers to communicate data with the frontend and it uses views as methods and viewsets as classes which were difficult to comprehend from scratch.

6.6 User Authentication and Security

Implementing a secure and reliable user authentication system with password encryption, password recovery mechanisms, and secure session management is crucial to protect user data and prevent unauthorized access.

6.7 Database Management

Efficiently organizing and managing the extensive amount of country-related data, including general facts, history, culture, sports, forms of government, tourist destinations, and ocean/sea information, requires a well-structured and scalable database design.

6.8 Web Scraping for Country Information

Extracting and updating real-time country information through web scraping can be challenging due to the dynamic nature of websites and potential changes in the HTML structure. Ensuring the reliability and accuracy of the scraped data is essential.

6.9 Interactive Map Integration

Integrating interactive maps with markers and click events to display specific country information or ocean/sea details requires using appropriate mapping libraries and APIs. Ensuring seamless integration and smooth user experience across different devices and browsers is a challenge.

6.10 Quiz Functionality

Designing and implementing the quiz module, including multiple-choice questions and gaming quizzes, requires creating a robust question database, managing user progress, calculating scores, and tracking user levels and XP. Ensuring the quiz logic is error-free and handling concurrent user interactions can be complex.

6.11 Real-Time Chat System

Developing a real-time chat system that allows users from different countries to communicate requires using technologies such as websockets or long-polling to enable instant message delivery. Handling concurrent connections, message broadcasting, and maintaining message history can be challenging.

6.12 User Interface and User Experience

Designing an intuitive and visually appealing user interface that provides easy navigation, responsive layout, and consistent user experience across different devices and screen sizes requires careful consideration of usability principles and responsive design techniques.

6.13 Performance Optimization

Optimizing the website's performance, such as minimizing load times, efficient caching mechanisms, and handling concurrent user requests, is crucial to provide a seamless and responsive user experience, especially when dealing with large amounts of data and interactive features.

6.14 Scalability and Traffic Management

Designing the website to handle increased traffic, user registrations, quiz participation, and chat interactions requires building a scalable and resilient architecture that can accommodate growth and efficiently handle concurrent requests without compromising performance.

6.15 Testing and Quality Assurance

Conducting thorough testing, including unit testing, integration testing, and user acceptance testing, to identify and fix bugs, ensure functionality, and validate user experience is essential to deliver a stable and reliable website.

CHAPTER 7

FUTURE WORK & CONCLUSION of Mapping The World

7.1 Future Work

Our web based application has a strong basis to establish it as an independent product in the market. To do so, there's a few features we would like to add-

1. User Profiles: Implementing a comprehensive user profile system where users can personalize their accounts, add profile pictures, and provide additional information about themselves.

2. Social Media Integration: Allowing users to connect their social media accounts to the website, enabling them to share their quiz scores, achievements, or interesting country facts with their social network.

3. Language Localization: Adding support for multiple languages to cater to users from different regions, enhancing accessibility and usability.

4. User-Generated Content: Introducing features that allow users to contribute their own country-related content, such as writing articles, submitting photos, or sharing personal travel experiences.

5. Advanced Quiz Modes: Introducing more challenging quiz modes, such as timed quizzes, multiplayer quizzes, or themed quizzes based on specific topics like geography, history, or culture.

6. Gamification Elements: Implementing gamification elements like badges, achievements, and leaderboards to enhance user engagement and motivation.

7. Mobile Application: Developing a mobile application for iOS and Android platforms to provide a seamless and optimized experience for users on their mobile devices.

8. Virtual Reality (VR) Integration: Integrating VR technology to provide users with immersive experiences, such as virtual tours of famous landmarks or interactive 360-degree views of countries.

9. Augmented Reality (AR) Features: Implementing AR functionality to overlay information about countries or landmarks when users point their device's camera at specific locations on the map.

10. Enhanced Chat Features: Adding features like group chats, voice messaging, or video calls to enrich the communication experience among users.

11. User Feedback and Ratings: Implementing a feedback system where users can provide ratings and reviews for countries, tourist destinations, or quiz questions, helping other users make informed decisions.

12. Machine Learning Integration: Utilizing machine learning algorithms to provide personalized recommendations for countries to visit, quizzes to play, or chat connections based on user preferences and behavior.

13. Integration with External APIs: Integrating with external APIs to provide real-time data on weather, currency exchange rates, or current events in different countries, enhancing the website's informational value.

14. Social Sharing: Enabling users to share interesting country facts, quiz results, or chat experiences directly on their social media platforms.

15. User Forums: Creating discussion forums or community boards where users can ask questions, engage in country-related discussions, or seek travel advice from other users.

16. Interactive Map Customization: Allowing users to customize the map display, such as choosing different map themes, toggling layers, or adding personal markers for visited countries.

17. Advanced Search Functionality: Implementing advanced search features to allow users to search for specific country information, quiz topics, or chat connections based on various criteria.

18. User Notifications: Introducing push notifications or email notifications to keep users updated about new quiz rounds, leaderboard updates, or chat requests.

19. Data Analytics and Insights: Incorporating analytics tools to gather user behavior data, perform data analysis, and gain insights into user preferences, popular quiz topics, or frequently accessed country information.

20. Continuous Content Expansion: Regularly updating and expanding the database of country information, quiz questions, and chat features to keep the website fresh, engaging, and informative for users.

By pursuing these future developments, the "Mapping the World" website can evolve into a comprehensive and dynamic platform, providing users with an immersive, educational, and interactive experience related to countries and cultures worldwide.

7.2 Key Learnings from Building the 'Mapping the World' Website

Based on the provided information about the "Mapping the World" website and the challenges and future work associated with it, here are some key takeaways of what we have learned:

1. Building a Comprehensive Website: We have gained an understanding of the various modules and features involved in developing a complex website that encompasses user authentication, country information, quizzes, and chat functionality.

2. User Registration and Login: We have learned how to implement a user registration system and login mechanism, including password recovery options to enhance the user experience and security.

3. Database Management: We have gained insights into organizing and managing a database to store and retrieve extensive country-related information, including general facts, history, culture, and more.

4. Web Scraping: We have learned how to extract and update real-time country information using web scraping techniques, ensuring the accuracy and reliability of the scraped data.

5. Interactive Map Integration: We have understood the process of integrating interactive maps with markers and click events to display specific country information, providing you with an engaging user experience.

6. Quiz Functionality: We have learned how to design and implement quiz features, including multiple-choice questions and gaming quizzes, managing your progress, calculating scores, and implementing leaderboard functionality.

7. Real-Time Chat System: We have gained knowledge about developing a real-time chat system that enables you and other users from different countries to communicate instantly, utilizing technologies like websockets or long-polling.

8. User Interface and User Experience: We have understood the importance of designing an intuitive and visually appealing user interface, ensuring a responsive layout and a consistent user experience across different devices.

9. Performance Optimization and Scalability: We have learned about optimizing website performance, implementing caching mechanisms, and designing a scalable architecture to handle increased traffic and your interactions.

10. Future Development Considerations: We have explored potential future enhancements for the website, such as user profiles, social media integration, advanced quiz modes, mobile application development, and machine learning integration.

These learnings provide us with a foundation for developing a feature-rich and interactive website, with a focus on user engagement, content management, and providing a seamless user experience.

7.2 Conclusion

In conclusion, the development of the "Mapping the World" website presents an exciting opportunity to create an interactive platform for users to explore and learn about countries worldwide. Throughout the process, you have gained valuable insights into various aspects of website development, including user authentication, country information, quizzes, and chat functionality.

One of the key learnings from this project is the importance of user registration and login systems. By implementing these features, you have enabled both administrators and users to access the website's functionalities. Administrators have the authority to manage the application and make necessary changes, while users can register, login, and access personalized features.

Database management has played a crucial role in organizing and storing extensive country-related information. By leveraging database systems, you have been able to store and retrieve data efficiently, ensuring that users can access general facts, historical events, cultural insights, and sports-related information about various countries.

Web scraping has proved to be a valuable technique for obtaining real-time country information. By utilizing web scraping methods, you have ensured that the website's content

remains up-to-date, providing users with accurate and relevant information about countries worldwide.

The integration of interactive maps has significantly enhanced user experience. Through the use of markers and click events, users can explore specific countries, view their general facts, history, culture, and sports achievements. The map feature has made learning about countries more engaging and interactive.

The quiz functionality has added an element of gamification to the website, enabling users to test their knowledge about different countries. The implementation of multiple-choice questions and gaming quizzes has provided users with a fun and challenging experience. Additionally, the inclusion of a leaderboard has fostered healthy competition among users, motivating them to improve their scores and strive for the top position.

The real-time chat system has facilitated global communication and interaction among users. By allowing users from different countries to send chat requests and exchange messages, the website has fostered a sense of community and connectedness. Users can share their experiences, ask questions, and engage in discussions about countries, creating a collaborative learning environment.

Looking towards the future, there are numerous opportunities for further development and enhancement of the website. Implementing user profiles would allow users to personalize their accounts, fostering a sense of identity and community within the platform. Integrating social media features would enable users to share their achievements, quiz scores, and interesting country facts with their social networks, expanding the reach of the website.

Additionally, language localization would enhance accessibility by providing support for multiple languages, catering to users from different regions. Developing a mobile application would provide users with a seamless and optimized experience on their mobile devices, increasing accessibility and convenience.

The incorporation of advanced technologies such as virtual reality (VR) and augmented reality (AR) would offer users immersive experiences, enabling them to explore countries and landmarks virtually. Machine learning integration could personalize the user experience by providing tailored recommendations for countries to visit, quizzes to play, and chat connections based on user preferences and behavior.

Continuous content expansion and updates would ensure that the website remains fresh, informative, and engaging for users. Regularly adding new country information, quiz questions, and chat features would keep users coming back for new experiences and knowledge.

In conclusion, the development of the "Mapping the World" website has provided you with a comprehensive understanding of website development, user experience, and content management. It has empowered you to create an interactive platform where users can explore countries, test their knowledge, and connect with people from around the world. By embracing future opportunities and advancements, the website has the potential to become a go-to resource for individuals seeking to learn, engage, and connect with the diverse cultures and countries that make up our world.

Reference [new page]

REST API

React Router Version Updates

Three Tier Architecture

FastAPI Docs

SqLite