

Project Document

for

TeamSync

Prepared by :

Team Members:

Anshul Ranjan - PES1UG21CS096

Aradhana Mahati - PES1UG21CS105

Ashlin Furtado - PES1UG21CS119

Ashwin Ragupathy - PES1UG21CS123

PES University, RR Campus

15 November 2023

Table of Contents

1. <i>Proposal of the Project</i>	3
2. <i>Software Requirements Specification</i>	5
3. <i>Project Plan</i>	23
4. <i>Design Diagram</i>	25
5. <i>Test Cases</i>	31
6. <i>Screenshots of Output</i>	36

Proposal of the Project

TeamSync

PES University, RR Campus

1 September 2023

1. Project Proposal

Project Title: TeamSync

Team Members:

Aradhana Mahati	-	PES1UG21CS105
Anshul Ranjan	-	PES1UG21CS096
Ashlin Furtado	-	PES1UG21CS119
Ashwin Ragupathy	-	PES1UG21CS123

Synopsis:

Project Objective:

The primary objective of the TeamSync project is to develop a web application that facilitates the formation of effective teams for capstone projects and hackathons in our campus. TeamSync aims to streamline the team-building process, enhance collaboration, and increase participation in these educational and competitive events.

Key Features and Components:

1. User Authentication and Profiles:

- TeamSync offers user authentication and profile creation functionality.
- Users can securely log in, create profiles, and provide essential information, including skills, interests, and past project experiences.

2. Recommender System:

- The core feature of TeamSync is its recommender system.
- Using user profiles, the system recommends potential team members based on skill compatibility and project alignment.

3. User Interface:

- TeamSync has an intuitive and user-friendly frontend design.
- The interface simplifies the user experience, making team formation and project planning effortless.

4. Messaging System:

- TeamSync includes a built-in messaging system to enable users to initiate and maintain conversations with potential team members.

5. Backend :

- Behind the scenes, TeamSync operates on a robust and scalable backend infrastructure.
- This backend securely manages user data, team information, and system performance, ensuring a smooth user experience.

Benefits:

- Increased Efficiency: TeamSync reduces the time and effort required to form teams for capstone projects and hackathons, enhancing overall efficiency.
- Quality Team Matches: The recommender system improves the quality of team matches by considering user skills and preferences, resulting in more productive and easier collaborations.
- Enhanced Participation: TeamSync encourages more students to participate in hackathons, fostering innovation and learning within the college community.
- Effective Collaboration: The integrated messaging system facilitates efficient communication and project coordination, ensuring that teams stay focused and deliver exceptional results.

TeamSync aims to empower individuals and teams to excel in capstone projects and hackathons, ultimately contributing to the growth and development of students.

2. Software Requirements Specification

for

TeamSync

Version 1.0 approved

Prepared by Project Managers:

Team Members:

Anshul Ranjan - PES1UG21CS096

Aradhana Mahati - PES1UG21CS105

Ashlin Furtado - PES1UG21CS119

Ashwin Ragupathy - PES1UG21CS123

PES University, RR Campus

16 October 2023

Table of Contents

Revision History	2
1. Introduction	3
1.1 Purpose	3
1.2 Intended Audience.....	3
1.3 Product Scope	3
1.4 References	3
2. Overall Description	4
2.1 Product Perspective.....	4
2.2 Product Functions	4
2.3 User Classes and Characteristics	4
2.4 Operating Environment	4
2.5 Design and Implementation Constraints	5
2.6 User Documentation	5
2.7 Assumptions and Dependencies	5
3. External Interface Requirements	6
3.1 User Interface	6
3.2 Software Interface.....	6
3.3 Communication Interface.....	7
4. Analysis Models	8
5. System Features.....	9
5.1 Landing Page	9
5.1.1 Description and Priority	9
5.1.2 Stimulus/Response Sequences.....	9
5.1.3 Functional Requirements	9
5.2 Login Page	9
5.2.1 Description and Priority	9
5.2.3 Functional Requirements	9
5.3 Signup Page	10
5.3.1 Description and Priority	10
5.4 Student Details Input Form / Event Input Form.....	11
5.5 Navigation Bar	12
5.6 Main Page.....	13
6. Other Non Functional Requirements	15
Performance Requirements.....	15
7. Appendix.....	17

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

TeamSync is a website designed for students of PES University to connect with other students with similar interests in terms of working on projects. Specifically, the purpose of TeamSync is for students to find teammates for their Capstone and Hackathon projects. This platform will serve as a centralized hub where people can connect, share ideas and collaborate on projects that align with their skills, goals and interests. Often, students find it difficult to find teammates for projects. TeamSync aims to make the process of finding teammates easier as well as more thorough.

1.2 Intended Audience

TeamSync is designed keeping the interests of students in mind. It is designed for students, by students. With restrictions on parameters like team size, diverse CGPA requirements as well as gender diversity requirements, students find it hard to form teams that not only fit the criteria, but also have similar interest and levels of commitment to the project.

1.3 Product Scope

The primary objectives and purposes of this project can be summarized as -

- Facilitating Collaboration: The main goal of the website is to enable individuals to find like-minded individuals or teams with complementary skills and interests for project collaboration. Users should be able to easily connect with others who share their project goals.
- Networking and Community Building: The website aims to foster a sense of community and networking among users. It should provide features that allow users to build connections, communicate, and exchange ideas with others who have similar project aspirations.
- Project Discovery: Users should be able to discover new projects or opportunities that match their interests, skills, and goals. The platform should offer project listings and recommendations to help users find relevant opportunities.
- Skill Matching: The website should have mechanisms for users to list their skills, expertise, and project interests. It should then use this information to match users with suitable projects and collaborators.
- User Engagement and Interaction: Encouraging user engagement and interaction is crucial. The website should offer features like forums, messaging, and commenting to facilitate communication and discussion among users.

1.4 References

To Be Filled

2. Overall Description

2.1 Product Perspective

TeamSync is a dynamic platform, whether accessed through a website or app, conceived to connect individuals with shared project aspirations, skills, and interests. It is a specialized solution catering to the collaborative needs of users, focusing on team formation for hackathons, projects, events, and competitions. The primary objectives encompass facilitating collaboration, nurturing community bonds, aiding project discovery, skill matching, and enhancing user interaction. TeamSync seamlessly integrates into users' workflows, enhancing their ability to connect, find projects, and engage within a supportive community. With an intuitive user-centric design, TeamSync stands as a pivotal tool in the realm of team collaboration and project development.

2.2 Product Functions

TeamSync offers a range of essential functions to create a robust platform for seamless team collaboration and community building. These functions include user registration and profile creation, enabling individuals to list their skills and project interests. It provides a project discovery feature, which offers project listings and recommendations based on user preferences. Skill matching algorithms facilitate the pairing of users with relevant projects and potential collaborators. TeamSync also incorporates interactive features such as forums, messaging, and commenting for users to engage in meaningful discussions and communication. Furthermore, it maintains a database of user profiles and projects, ensuring efficient search and connection capabilities. These functions collectively empower users to connect with like-minded individuals, discover project opportunities, and foster a sense of community while pursuing their project goal.

2.3 User Classes and Characteristics

User Class : Administrator

1. Administrator can Monitor the data and web page
2. Administrator can modify/add/update the details in the Database.
3. Administrator can define the team requirements.

User Class : Team Leader

1. Team Leader creates a new team, along with the team requirements and project details.
2. Team Leader also can access student's info and accordingly accept or reject that particular student.

User Class : Students

1. Student can search for teams , that are compatible to his/her skillsets.

2.4 Operating Environment

- OE – 1 : Web Interface
- The software will primarily operate through a web browser interface , providing Students, Team Leaders an Interactive Dashboard.
- The application will be implemented using the MERN Stack.
- This web based application will facilitate easy collaborations.

2.5 Design and Implementation Constraints

Will be updated in the next revision after design plan.

2.6 User Documentation

- TeamSync will offer an HTML-based hierarchical and cross-referenced online help system that comprehensively outlines all system functions.
- User documentation will commence with an introductory section, providing an overview of the TeamSync platform and identifying its target audience.
- The documentation will encompass subjects like initial system configuration, user roles, fundamental and advanced operations, problem resolution, recommended procedures, and security guidelines

2.7 Assumptions and Dependencies

Assumptions:

1. **(AS-1)Internet Connectivity:** It is assumed that users of TeamSync have reliable internet connectivity to access the platform and its features.
2. **(AS-2)Web Browsers:** Users are expected to use modern web browsers compatible with the application, as mentioned in the operating environment.
3. **(AS-3)User Authenticity:** It is assumed that users will provide accurate and truthful information when creating profiles and interacting on the platform.

Dependencies:

1. **(DE-1)Database Infrastructure:** The application relies on MongoDB as its primary database. The database infrastructure must be maintained and available for data storage and retrieval.
2. **(DE-2)Maintenance and Support:** To ensure the continued operation and improvement of TeamSync, there is a dependency on ongoing maintenance and support from the

3. External Interface Requirements

3.1 User Interface

3.1.1 Login Page (UI-1)

- Standard fields for username/email and password.
- "Forgot Password" option for password recovery

3.1.2 Registration Page (UI-2)

- Fields for name, email, password, and possibly profile picture upload.
- Validation for email format and password strength.

3.1.3 User Profile Page (UI-3)

- Fields for users to enter and update their information.
- Options to upload, change, or remove a profile picture.
- Options to upload, edit, or remove their skill sets
- Save and Cancel buttons for editing.

3.1.4 Dashboard / Homepage (UI-4)

- Personalized content based on user activity.
- Quick access to key features like team recommendations, messaging, and user profile.

3.1.5 Navigation Menu (UI-5)

- Easily accessible menu for navigation to different sections.
- Highlighting the current section or page.
- Logout option for logging out.

3.2 Software Interface

3.2.1 Web Application (TeamSync) and User Databases(SI-1)

- Database Software: A database management system - MongoDB for storing user data and profiles.
- Data Flow: The web application sends user data (user profiles, login information, etc.) to the user database for storage. It retrieves user profiles and other relevant data for display within the application.

3.2.2 Frontend Libraries(SI-2)

- Frontend Libraries: HTML, CSS, JavaScript, and a JavaScript framework React.
- Data Flow: The web application uses these libraries and tools to render the user interface and provide an interactive user experience.

3.2.3 Operating System and Hosting Platform(SI-3)

- Operating System: The OS used for hosting the application - Windows.

- Data Flow: The operating system hosts your web application and backend, ensuring the server's functionality.

3.2.4 Data Sharing Mechanism(SI-4)

- Implementation constraints related to data sharing, such as the use of tokens for authentication, the encryption of sensitive data, and the use of a global data area in the backend for managing real-time communication.

3.3 Communication Interface

3.3.1 User Authentication and Authorization(CI-1)

- Protocol: HTTPS for secure communication.
- Message Formatting: JSON or XML for authentication and authorization requests.
- Security: Implement secure authentication mechanisms like OAuth 2.0 or token-based authentication.
- Data Transfer Rates: Ensure speedy authentication and authorization processes to minimize login time.

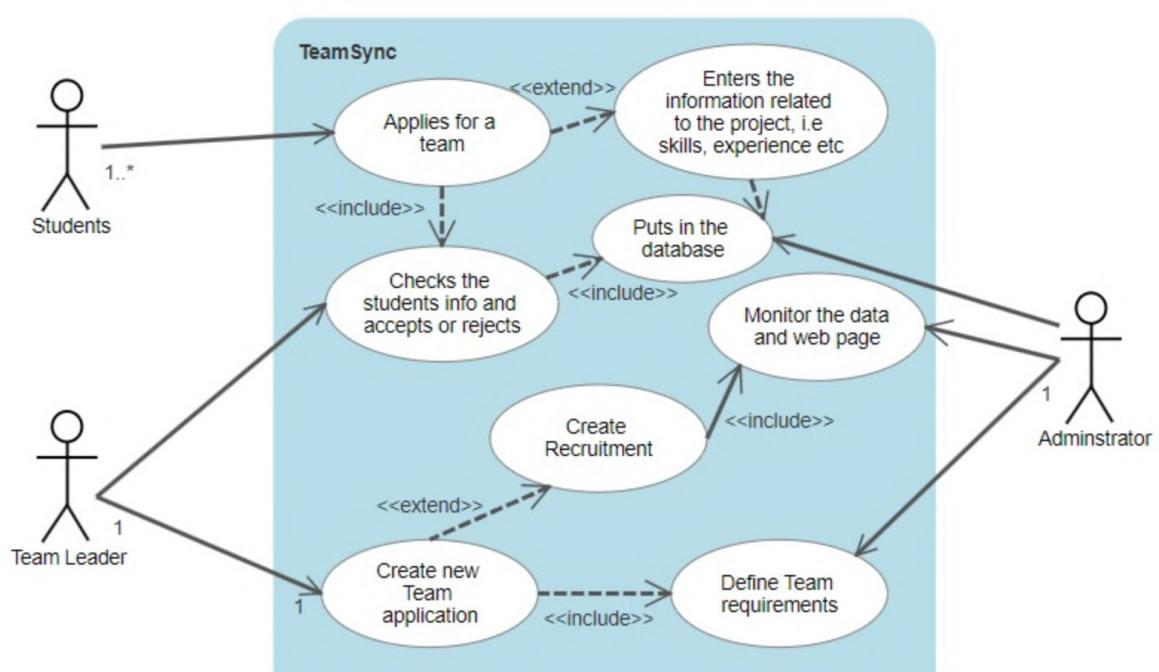
3.3.2 Database Communication(CI-2)

- Database Queries: Use SQL or NoSQL queries to retrieve and update user profile data.
- Message Formatting: Data exchanged with the database should be in a format that the database system understands (e.g., SQL statements or JSON for NoSQL).
- Security: Implement database access controls to restrict unauthorized access to user data.
- Data Transfer Rates: Database queries should be optimized for performance to prevent latency in user profile retrieval.

3.3.3 Synchronization Mechanisms(CI-3)

- Implement mechanisms to ensure data consistency and synchronization, particularly for real-time messaging and user data updates.

4. Analysis Models



5. System Features

5.1 Landing Page

5.1.1 Description and Priority

- **Description:** The Landing Page with Login and Sign-Up Buttons serves as the primary entry point for users visiting the website or application. It presents a visually appealing and informative interface that guides users to perform key actions, including logging in to existing accounts or creating new ones.
- **Priority:** Medium

5.1.2 Stimulus/Response Sequences

- **User Action 1:** The user navigates to the website's main URL.
- **System Response 1:** The Landing Page is displayed, showcasing essential information about the website or application.
- **User Action 2:** The user clicks on the "Login" button.
- **System Response 2:** The system redirects the user to the Login Page, prompting them to enter their login credentials.
- **User Action 3:** The user clicks on the "Sign Up" button.
- **System Response 3:** The system directs the user to the Registration Page, allowing them to create a new account.

5.1.3 Functional Requirements

- **REQ-1:** Landing Page Design.
The Landing Page will feature an attractive and user-friendly design. It will contain text, images, or graphics that provide a brief overview of the website's purpose and value.
- **REQ-2:** "Login" Button
Clicking the "Login" button shall direct the user to the Login Page.
- **REQ-3:** "Sign Up" Button.
Clicking the "Sign Up" button shall direct the user to the Registration Page.

5.2 Login Page

5.2.1 Description and Priority

- **Description:** The Login Page is responsible for authenticating users and granting access to the website's services. It allows users to log in with their credentials.
- **Priority:** High

5.2.2 Stimulus/Response Sequences

- **User Action 1:** User enters their valid credentials and clicks the "Login" button.
- **System Response 1:** The system verifies the credentials. If they are correct, the user is granted access to their account and redirected to the main page. If the credentials are incorrect, an error message is displayed.
- **User Action 2:** User clicks on the "Forgot Password" link.
- **System Response 2:** The system displays a form for the user to enter their email address. After submission, the user receives an email with a password reset link.

5.2.3 Functional Requirements

- **REQ-1:** User Authentication
The Login Page shall provide a form with fields for entering a username and password. The system shall validate the user's credentials against the database. If the user's credentials are correct, they shall be granted

- access to their account. If the user's credentials are incorrect, an error message shall be displayed
- **REQ-2:** Forgot Password.
The Login Page shall include a "Forgot Password" link. When the user clicks this link, a form shall be displayed to enter their email address. After submitting the email address, the system shall send a password reset link to that email address.

5.3 Signup Page

5.3.1 Description and Priority

- **Description:** The Sign-Up Page allows new users to create accounts on the website, providing essential user information, including a unique username, email address, and a secure password.
- **Priority:** Medium

5.3.2 Stimulus/Response Sequences

- **User Action 1:** The user enters their desired username, email address, and password, then clicks the "Sign Up" button.
- **System Response 1:** The system validates the provided information. If validation is successful, a new user account is created, and the user is redirected to the Login Page. If validation fails, error messages are displayed to guide the user.
- **User Action 2:** The user receives a confirmation email and clicks on the provided link to verify their account.
- **System Response 2:** The user's account is marked as verified, allowing them to log in.

5.3.3 Functional Requirements

- **REQ-1:** User Input Fields

The Sign-Up Page shall include input fields for username, email address, and password.

Input fields shall have associated labels and placeholders for clarity.

Field labels shall be visually associated with the corresponding input fields.

- **REQ-2:** Input Validation

The system shall validate user input to ensure accuracy and adherence to the following rules:

Username: Must be unique, alphanumeric, and within a specified character limit.

Email Address: Must be in a valid email format.

Password: Must meet minimum complexity requirements (e.g., minimum length, inclusion of special characters, numbers, and letters).

- **REQ-3:** User Registration

Upon successful input validation, the system shall create a new user account. The system shall assign a unique user ID and securely hash the user's password before storing it.

- **REQ-4:** Redirect to Login Page

After successful registration, the system shall redirect the user to the Login Page, allowing them to log in with their newly created account.

5.4 Student Details Input Form / Event Input Form

5.4.1 Description and Priority

- **Description:** The Student Details Input Form is a web-based form that allows students to input and submit their personal and academic information, including SRN, year, branch, skills, and projects completed.
- **Priority:** Medium

5.4.2 Stimulus/Response Sequences:

- **User Action 1:** The user navigates to the Student Details Input Form.
- **System Response 1:** The form is displayed, ready for the user to enter and submit their details.
- **User Action 2:** The user enters their SRN, selects their year, branch, and lists their skills and projects completed. Afterward, they click the "Submit" button.
- **System Response 2:** The system validates the provided information. If validation is successful, the data is saved. If validation fails, error messages are displayed to guide the user in making corrections.

5.4.3 Functional Requirements

- **REQ-1:** Form Structure

The Student Details Input Form shall include input fields for the following:

SRN (Student Registration Number)

Year (drop-down selection for year of study)

Branch (drop-down selection for the student's academic branch)

Skills (a text field where the student can list their skills, separated by commas or semicolons)

Projects Done (a text field where the student can list completed projects)

- **REQ-2:** Input Validation

The system shall validate user input to ensure accuracy and adherence to specific criteria:

SRN: Must be unique and match the format specified by the institution.

Year: Must be selected from the available options (e.g., 1st year, 2nd year).

Branch: Must be selected from the available options (e.g., Computer Science, Electrical Engineering).

Skills: May contain alphanumeric characters and separators (e.g., commas, semicolons), with no leading or trailing separators.

Projects Done: May contain alphanumeric characters, punctuation, and line breaks.

- **REQ-3:** Data Storage

Upon successful validation of user input, the system shall save the provided information in a database.

SRN, Year, Branch, Skills, and Projects Done shall be associated with the respective student's profile.

- **REQ-4:** Error Handling

The Student Details Input Form shall provide user-friendly error messages in case of invalid information entry.

Error messages shall be displayed near the respective input fields and guide users in correcting their input.

- **REQ-5:** Confirmation Message

After successful submission, the system shall display a confirmation message to the user, indicating that their details have been successfully recorded.

5.5 Navigation Bar

5.5.1 Description and Priority

- **Description:** The Navigation Bar component is a crucial part of the application's user interface, offering easy access to different sections and components. It allows users to seamlessly navigate between features like the Login Page, Event Details Input Form, Student Details Input Form, Student Details Display Component, Event Display Component with "Apply" button, Recommended Students for Team, and Requests Display Component.
- **Priority:** High

5.5.2 Functional Requirements

- **REQ-1:** Access Control
The Navigation Bar shall be accessible to users upon login or authentication to the application.
- **REQ-2:** Main Page Access
It shall include a link or button that directs users to the application's main page, where they can access all integrated components.
- **REQ-3:** Event Details Input Form Access
The Navigation Bar shall provide a link or button for event organizers to access the Event Details Input Form for creating new events.
- **REQ-4:** Student Details Input Form Access
It shall include a link or button for students to access the Student Details Input Form for entering their information.
- **REQ-5:** Student Details Display Access
The Navigation Bar shall offer a link to access the Student Details Display Component, allowing users to view student profiles.
- **REQ-6:** Event Display Access
It shall include a link or button for users to access the Event Display Component with an "Apply" button, where they can view and apply for events.

- **REQ-7:** Recommended Students Access
The Navigation Bar shall provide a link for students to access the Recommended Students for Team feature, enabling them to specify requirements and preferences.

- **REQ-8:** Requests Display Access
It shall include a link or button that allows users to access the Requests Display Component, where they can manage incoming requests.
- **REQ-9:** User Profile Access
The Navigation Bar may offer a link for users to access their profile section, allowing them to view and edit their profile information.

- **REQ-10: Notifications Access**
It shall provide a notifications or alerts section that informs users of important events, such as incoming requests or application status.
- **REQ-11: Responsiveness**
The component shall be designed to be responsive and adapt to different screen sizes and devices, ensuring a user-friendly experience on both desktop and mobile platforms.

5.6 Main Page

5.6.1 Description and Priority

- **Description:** The Main Page serves as the central interface of the application, integrating various components for a seamless user experience. It includes components like the Login Page, Event Details Input Form, Student Details Input Form, Student Details Display Component, Event Display Component with "Apply" button, Recommended Students for Team, and Requests Display Component.
- **Priority:** High

5.6.2 Functional Requirements

- **REQ-1: User Authentication**
The Main Page shall include a Login Page component, which allows users to log in or authenticate to access their accounts.
- **REQ-2: Navigation**
After successful login, the Main Page shall provide a navigation mechanism for users to access different sections and components of the application.
- **REQ-3: Event Details Input Form**
The Main Page shall include the Event Details Input Form, allowing event organizers to input event information such as Event Name, Date, Time, Venue, Team Size, and Prize Money.
- **REQ-4: Student Details Input Form**
The Main Page shall include the Student Details Input Form, enabling students to input their details, such as SRN, Year, Branch, Skills, and Projects Done.
- **REQ-5: Student Details Display**
The Main Page shall include the Student Details Display Component, which retrieves and displays student details in a React card format.
- **REQ-6: Event Display with "Apply" Button**
The Main Page shall include the Event Display Component with an "Apply" button, allowing users to view and apply for events.
- **REQ-7: Recommended Students for Team**

The Main Page shall integrate the Recommended Students for Team feature, enabling students to specify requirements and preferences and receive recommendations for potential team members.

- **REQ-8:** Requests Display

The Main Page shall include the Requests Display Component, allowing users to manage incoming requests, such as connection requests and team invitations.

6. Other Non Functional Requirements

Performance Requirements

1. Response Time:

Requirement: The website should respond to user interactions (e.g., searching for projects, loading project listings, sending messages) within 2 seconds or less.

Rationale: Quick response times enhance user experience and engagement, making the platform more attractive to users. Users are more likely to stay and interact if the website is responsive.

2. Scalability:

Requirement: The website should be designed to handle a minimum of 1,000 concurrent users without a significant degradation in performance.

Rationale: Scalability is essential to accommodate a growing user base and ensure that the platform remains responsive during peak usage periods.

3. Database Query Performance:

Requirement: Database queries, such as searching for projects or users, should return results within 1 second for typical queries.

Rationale: Efficient database query performance is critical to ensure that users can quickly find projects or collaborators that match their criteria.

4. Content Delivery:

Requirement: Static content, such as images, stylesheets, and JavaScript files, should be delivered with a content delivery network (CDN) to reduce load times. These assets should load in less than 1 second.

Rationale: CDNs can significantly reduce load times, which is essential for a smooth user experience, especially for users accessing the platform from different geographical locations.

Safety Requirements

1. Data Privacy and Security:

Requirement: The system must adhere to applicable data protection laws (e.g., GDPR, CCPA) to safeguard user data and privacy. This includes secure storage of personal information, encrypted data transmission, and user consent for data collection.

Rationale: Protecting user data is essential to prevent unauthorized access, data breaches, and misuse of personal information.

2. User Authentication and Authorization:

Requirement: Access to sensitive user information and administrative functions must be restricted to authorized personnel only. User authentication and authorization mechanisms must be in place to prevent unauthorized access.

Rationale: Unauthorized access to sensitive data or system controls can lead to data breaches and misuse of the platform.

3. Content Moderation:

Requirement: Implement a content moderation system to identify and remove inappropriate, offensive, or harmful content. The system should prevent the dissemination of harmful or unsafe information.

Rationale: Content moderation is necessary to maintain a safe and respectful online environment.

4. User Guidelines and Policies:

Requirement: Users should be required to accept and adhere to a code of conduct and community guidelines. Violations of these guidelines should result in appropriate actions, such as warnings, suspensions, or bans.

Rationale: Clear guidelines promote safe and respectful interactions among users.

5. Compliance with Legal and Regulatory Requirements:

Requirement: Ensure compliance with all relevant legal and regulatory safety requirements, including those related to online platforms, data protection, and user safety.

Rationale: Non-compliance with legal and regulatory requirements can result in legal consequences and harm to users.

Security Requirements

1. Data Encryption:

Requirement: All data transmitted between the user's device and the server must be encrypted using industry-standard protocols, such as HTTPS for web traffic. Sensitive data at rest should also be encrypted.

Rationale: Encryption safeguards user data during transmission and while it's stored, reducing the risk of data breaches.

2. User Authentication:

Requirement: Users must be required to create accounts with strong passwords and use multi-factor authentication (MFA) for added security. Passwords must be securely stored using salted and hashed techniques.

Rationale: Strong user authentication practices reduce the risk of unauthorized access to accounts.

3. Access Control:

Requirement: Implement role-based access control (RBAC) to ensure that users have access only to the features and data relevant to their roles. Administrators should have the authority to manage access privileges.

Rationale: Access control limits the exposure of sensitive data to authorized personnel only.

4. Regular Security Audits:

Requirement: Conduct regular security audits and vulnerability assessments to identify and address potential security issues. Perform penetration testing to assess the system's resilience to attacks.

Rationale: Regular audits help identify and mitigate security vulnerabilities before they are exploited.

5. Protection Against Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF):

Requirement: Implement security measures to prevent and mitigate XSS and CSRF attacks. This includes input validation, output encoding, and anti-CSRF tokens.

Rationale: These measures protect against common web application security threats.

6. Data Privacy Compliance:

Requirement: Ensure compliance with data privacy regulations such as GDPR and CCPA, including user consent for data collection, processing, and the ability to provide users with their data upon request.

Rationale: Compliance with data privacy regulations is essential to protect user data and privacy.

7. Secure Session Management:

Requirement: Implement secure session management to prevent session fixation and session hijacking attacks. Session tokens should be unique, random, and securely managed.

Rationale: Secure session management is crucial to protect user sessions from unauthorized access.

8. Content Security Policies (CSP):

Requirement: Implement a content security policy to restrict the sources from which content can be loaded and executed. This prevents the execution of malicious scripts.

Rationale: CSP helps protect the platform from various types of attacks, including XSS.

9. Regular Software Patching:

Requirement: Ensure that all software components used in the system, including third-party libraries and frameworks, are regularly updated and patched to address security vulnerabilities.

Rationale: Regular patching is essential to mitigate known security vulnerabilities.

10. Privacy Certifications:

Requirement: If applicable, obtain and maintain privacy certifications, such as ISO 27001 or SOC 2, to demonstrate a commitment to user data privacy and security.

Rationale: Privacy certifications build trust with users and stakeholders, showcasing adherence to recognized privacy standards.

7. Appendix

Appendix A: Glossary

To be filled

Appendix B: Analysis Models

To be filled

Appendix C: To Be Determined List

To be fill

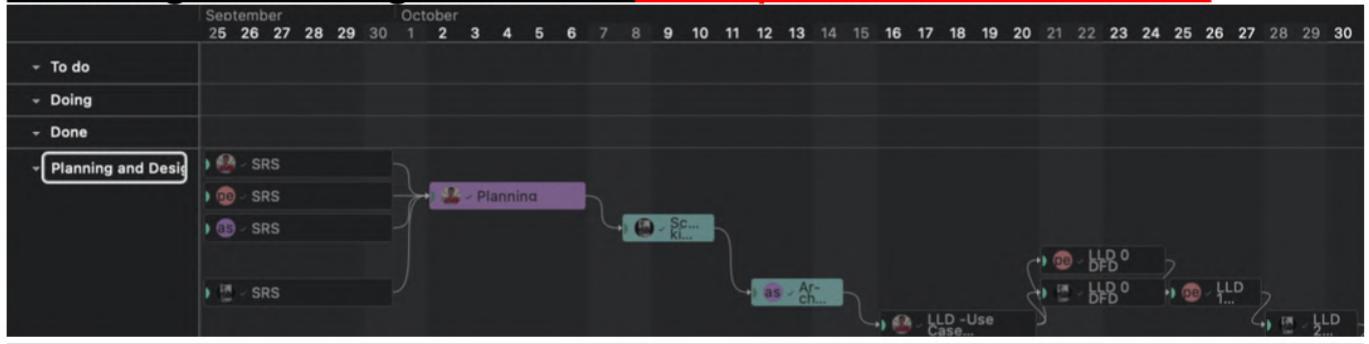
3. Project Plan for TeamSync

PES University, RR Campus

31 October 2023

Gantt Chart

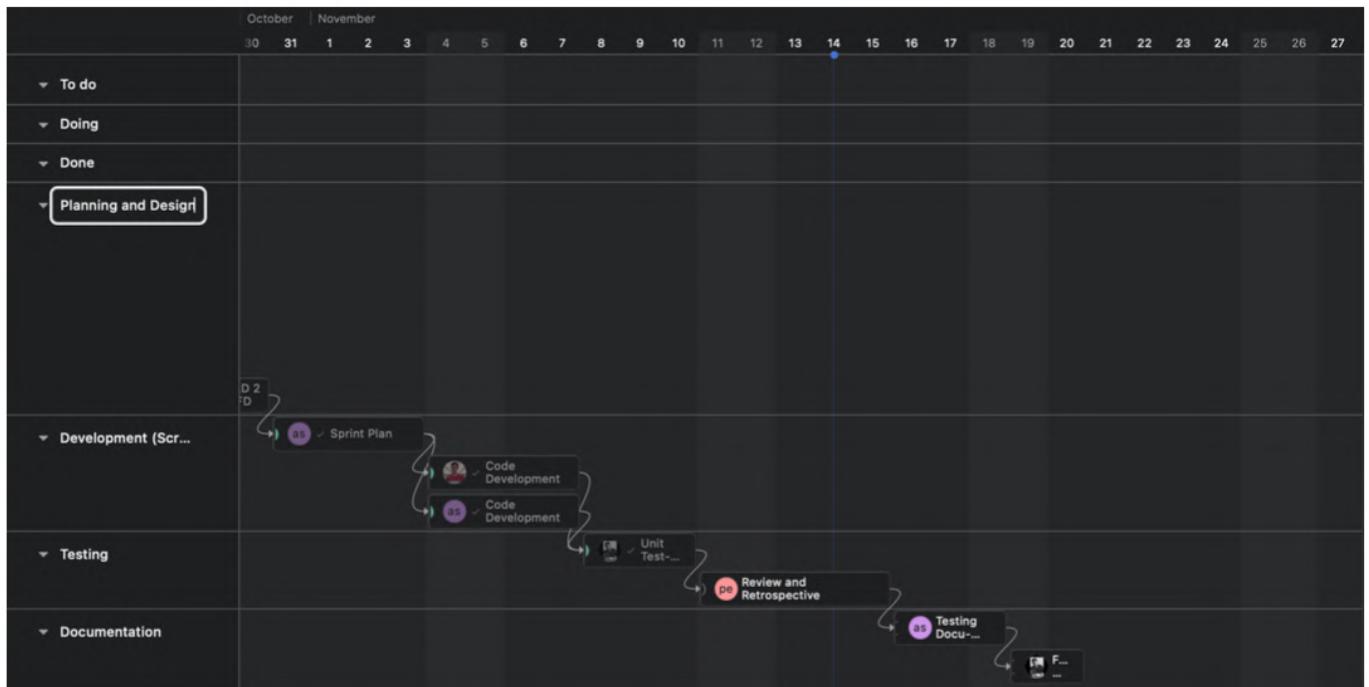
Planning and Design Phase – 25 September – 30 October



Development Phase – 31 October – 7 November

Testing Phase – 8 November – 15 November

Documentation Phase – 16 November – 21 November

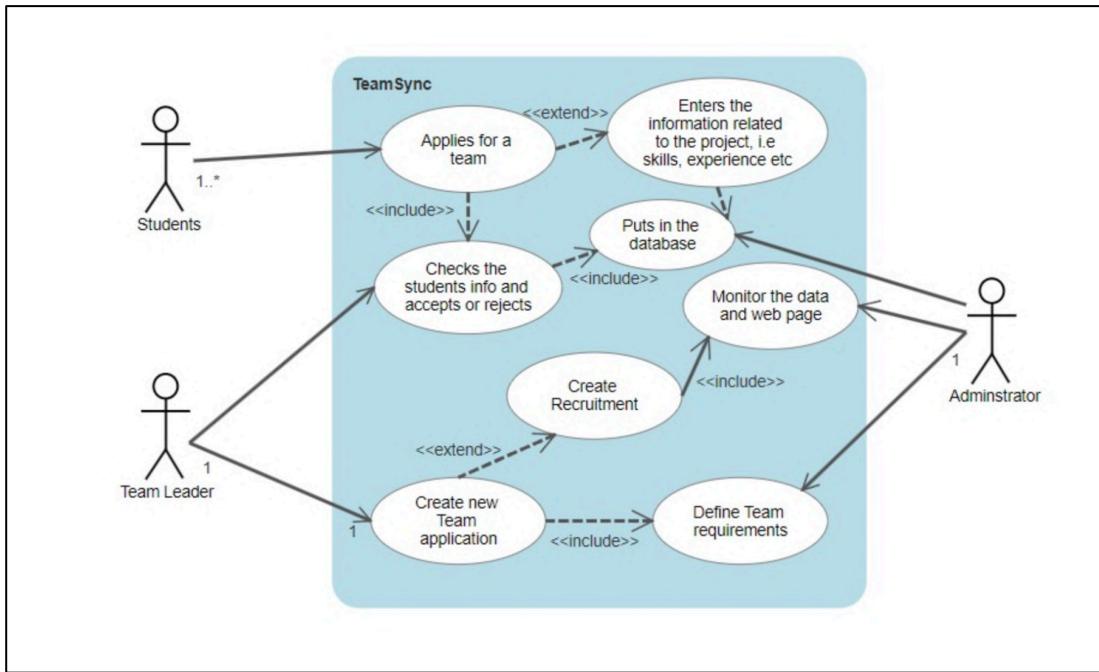


Design Diagram for TeamSync

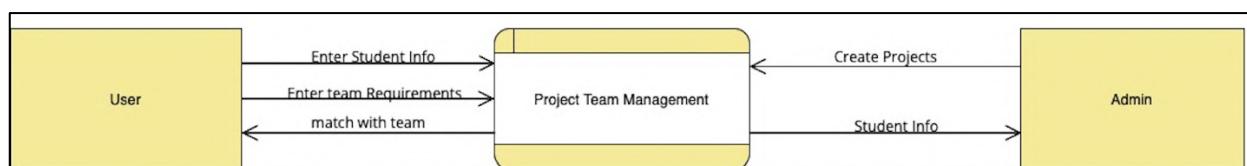
PES University, RR Campus

20 October 2023

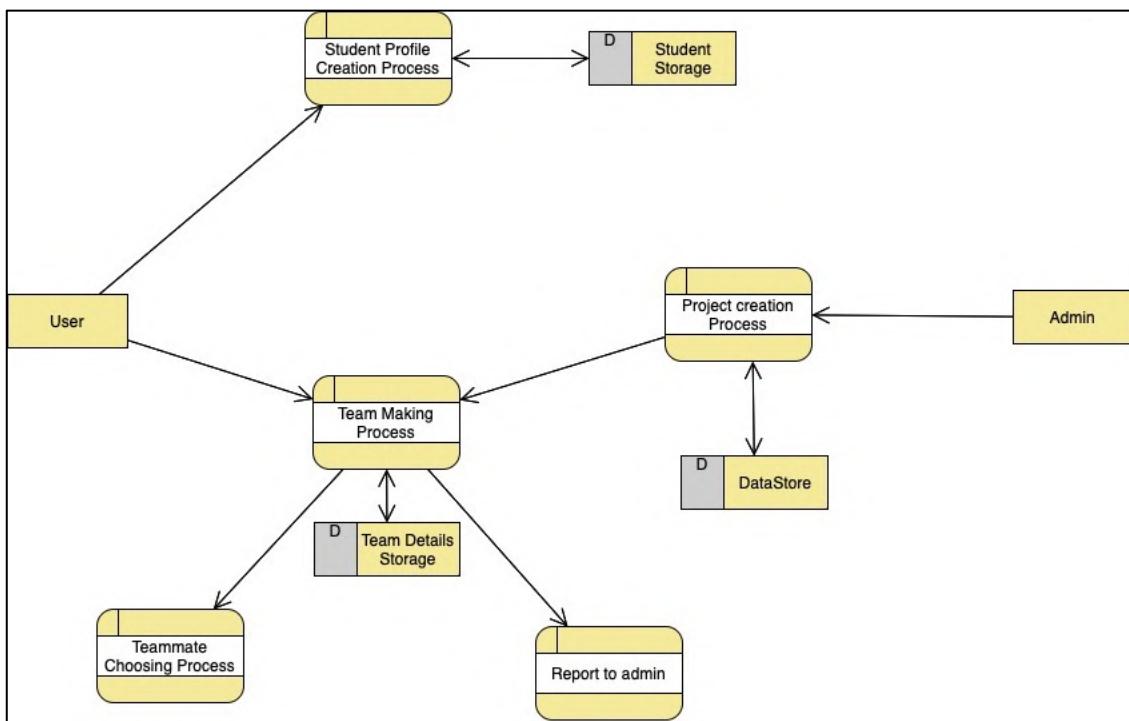
• Use Case Diagram



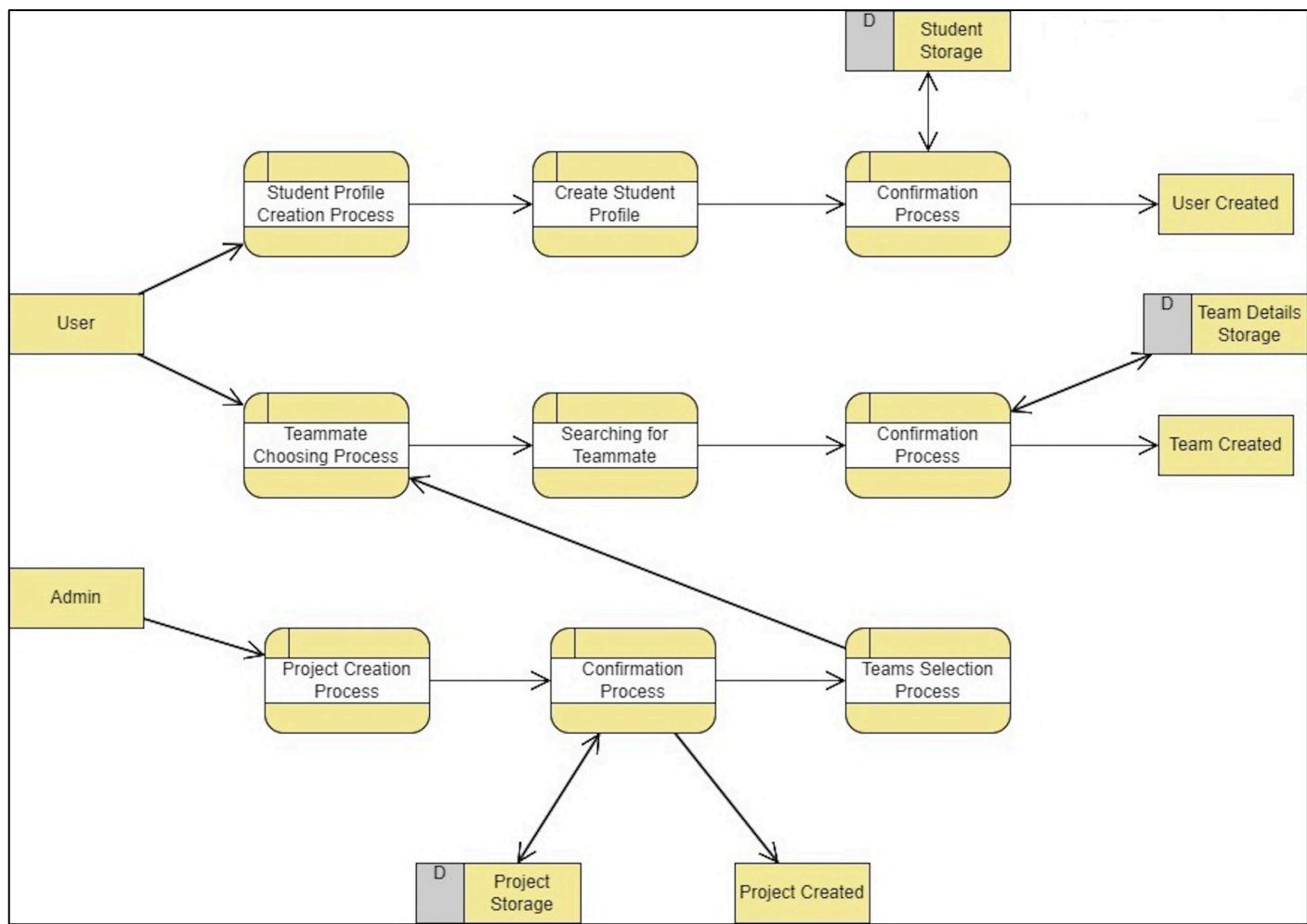
• Level 0 DFD



- Level 1 DFD



- Level 2 DFD



• Requirements Traceability Matrix

Sl. No	Requirement ID	Brief Description of Requirement	Architecture Reference	Design Reference	Code File Reference	Test Case ID	System Test Case ID
1	REQ-1	Landing Page Design	5.1	Screen Shots - 1	/pages/index.js	TC-01	STC-01
2	REQ-2	User Authentication	5.2	Level 2 DFD User	/pages/login.js	TC-02	STC-02
3	REQ-3	Input Validation	5.3	Level 1 DFD	/components/Auth/auth.js	TC-03	STC-03
4	REQ-4	Data Storage	5.1.2	Use Case diagram	/components/cards.js	TC-04	STC-04
5	REQ-5	Error Handling	6	Level 2 DFD	/pages/api/hello.js	TC-05	STC-05
6	REQ-6	Navigation to Main Page	5.6	Level 0 DFD	/pages/index.js	TC-06	STC-06
7	REQ-7	Form Structure	5.4	Screenshot – 4	/pages/team-form.js	TC-07	STC-07
8	REQ-8	Event and Student Form Access	5.4	Screenshot -5	/pages/need-teammate-form.js	TC-08	STC-08
9	REQ-9	Student Details Display Access	5.5	Screenshot - 3	/pages/teammates.js	TC-09	STC-09
10	REQ-10	Event Display Access	5.6	Screenshot - 2	/pages/teams.js	TC-10	STC-10
11	REQ-11	Notifications Access	5.1.2	Level 2 DFD	/pages/api/invite-email.js	TC-11	STC-11
12	REQ-12	Responsiveness	6	Screenshot - 6	/styles/globals.css	TC-12	STC-12

Implementation :

GitHub Link : <https://github.com/AnshulRanjan2004/TeamSync>

GitHub Repo:

The screenshot shows the GitHub repository page for 'TeamSync' (public). The main content area displays a list of commits from 'AnshulRanjan2004'. The commits are as follows:

File	Message	Time Ago
components	fixed the whole code	9 minutes ago
pages	fixed the whole code	9 minutes ago
public	initial commit	last week
styles	initial commit	last week
.env	fixed env	yesterday
.eslintrc.json	initial commit	last week
.gitignore	initial commit	last week
.prettierrc.js	initial commit	last week
LICENSE	Initial commit	last week
README.md	Initial commit	last week
apply-template.html	initial commit	last week

On the right side, there is an 'About' section with the following details:

- Team/Teammates finder app built with Next.js.
- Readme
- MIT license
- Activity
- 0 stars
- 1 watching
- 0 forks

Below the 'About' section are sections for 'Releases' (No releases published) and 'Packages' (No packages published).

Home Page:

The screenshot shows the VS Code editor with the 'TeamSync' project open. The left sidebar shows the file structure:

- EXPLORER
- TEAMSYNC
 - > .next
 - > components
 - > Auth
 - JS auth.js
 - JS authContext.js
 - JS appwrite.js
 - JS Card-need-teammate.js
 - JS Card.js
 - JS Faq.js
 - JS Features.js
 - JS Footer.js
 - JS Header.js
 - JS Layout.js
 - JS Navbar.js
 - > node_modules
 - > pages
 - > api
 - JS apply-email.js
 - JS hello.js
 - JS invite-email.js
 - JS _app.js
 - JS _document.js
 - JS choices.js
 - JS edit-team.js
 - JS index.js
 - JS login.js
 - JS need-teammate-form.js

Styling:

This screenshot shows the VS Code interface with the repository 'TeamSync' open. The left sidebar displays the project structure under 'TEAMSYNC'. The main editor area shows the content of the file '# globals.css'. The code uses Tailwind CSS utility classes and a media query for dark mode. The status bar at the bottom indicates the file was last modified 6 days ago.

```
You, 6 days ago | author (You)
1 @tailwind base;
2 @tailwind components;
3 @tailwind utilities;
4 @import "fonts.css";
5 @import url('https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=swap');

6 @layer base {
7   html {
8     font-family: 'Inter', 'Faro', 'Orkney', 'Montserrat', sans-serif;
9   }
10  h1 {
11    font-family: 'Faro', sans-serif;
12  }
13}

14 :root {
15   --foreground-rgb: 0, 0, 0;
16   --background-start-rgb: 214, 219, 220;
17   --background-end-rgb: 255, 255, 255;
18 }

19 @media (prefers-color-scheme: dark) {
20   :root {
21     --foreground-rgb: 255, 255, 255;
22     --background-start-rgb: 0, 0, 0;
23     --background-end-rgb: 0, 0, 0;
24   }
25 }

26 body {
27   color: rgb(var(--foreground-rgb));
28   background: linear-gradient(
29     to bottom,
30     var(--background-start-rgb),
31     var(--background-end-rgb)
32   );
33 }
```

Registration Form:

This screenshot shows the VS Code interface with the repository 'TeamSync' open. The left sidebar displays the project structure under 'TEAMSYNC'. The main editor area shows the content of the file 'team-form.js'. The code handles form submission, sets user ID, and performs validation. The status bar at the bottom indicates the file was last modified 6 days ago.

```
pages > JS team-form.js > Form
1 // Success
2 setUserId(response.id)
3
4 } catch (error) {
5   console.error('Error: ', error) // Failure
6   router.push('/')
7 }

8 fetchData()
9 , []

10 function SubmitForm(e) {
11   e.preventDefault()
12   const created_by = userId
13   const name = personName
14   const about = description
15   const available = true
16   const contact = contactEmail
17   const github_url = githubURL
18   const twitter_url = twitterURL
19   const skills = skills

20   // Check for missing fields
21   const missingFields = []
22   if (!name) missingFields.push('Hackathon Name')
23   if (!about) missingFields.push('Team Description')
24   if (!skills.length) missingFields.push('Team Skills')
25   if (!country) missingFields.push('Country')
26   if (!contactEmail) missingFields.push('Contact Email')
27   if (!githubURL) missingFields.push('GitHub URL')
28   // Display toast message if any field is missing
29   if (missingFields.length > 0) {
30     toast.error('Please fill all in the missing fields')
31     return
32   }
33 }
```

4. Test Cases for TeamSync

PES University, RR Campus

7 November 2023

UNIT TESTING

Unit Test Case ID	Requirement ID	Test Case	Expected Output	Expected Results	Status
UT -01	REQ-1	Load Landing Page	Landing Page displayed successfully	The landing page is visible with the designed layout	Pass
UT - 02	REQ-2	User Authentication	Successful login with valid credentials	User is redirected to the main page	Pass
UT - 03	REQ-3	Input Validation	Submit form with valid input	Form is accepted and processed successfully	Pass
UT - 04	REQ-4	Data Storage	Save data to the database	Frontend not connected to the Database for now	In Progress
UT - 05	REQ-5	Error Handling	Trigger an error condition	The error message is not displayed on the frontend	Fail
UT - 06	REQ-6	Navigation to Main Page	Click on the Main Page link	User is redirected to the main page	Pass
UT - 07	REQ-7	Form Structure	Fill out the form correctly	Form is structured as expected	Pass
UT - 08	REQ-8	Event and Student Form Access	Access event and student forms	Forms are accessible and display correctly	Pass
UT - 09	REQ-9	Student Details Display Access	Access student details display	Student details are displayed correctly	Pass
UT - 10	REQ-10	Event Display Access	Access event display	Events are displayed correctly	Pass
UT - 11	REQ-11	Notifications Access	Access notifications	Need special Paid APIs to send email notifications	Fail
UT - 12	REQ-12	Responsiveness	Check responsiveness on different devices	Application adapts to different screen sizes	Pass

INTEGRATION TESTING

Integration Test Case ID	Requirement IDs	Test Case	Expected Output	Expected Results	Status
IT - 01	REQ-1, REQ-6	Load Landing Page	Landing page components load without errors	Verify that images, styles, and links are displayed correctly	Pass
IT - 02	REQ-2, REQ-3	Login with Valid Credentials	Successful login and redirection	User is redirected to the main page after logging in	Pass
IT - 03	REQ-4, REQ-8	Save Data to Database through Form Submission	Data saved correctly in the database	Verify that user input is stored accurately in the database	In Progress
IT - 04	REQ-7, REQ-10	Access Event Display	Events displayed based on user input	Verify that events are displayed correctly according to user preferences	Pass

SYSTEM TESTING

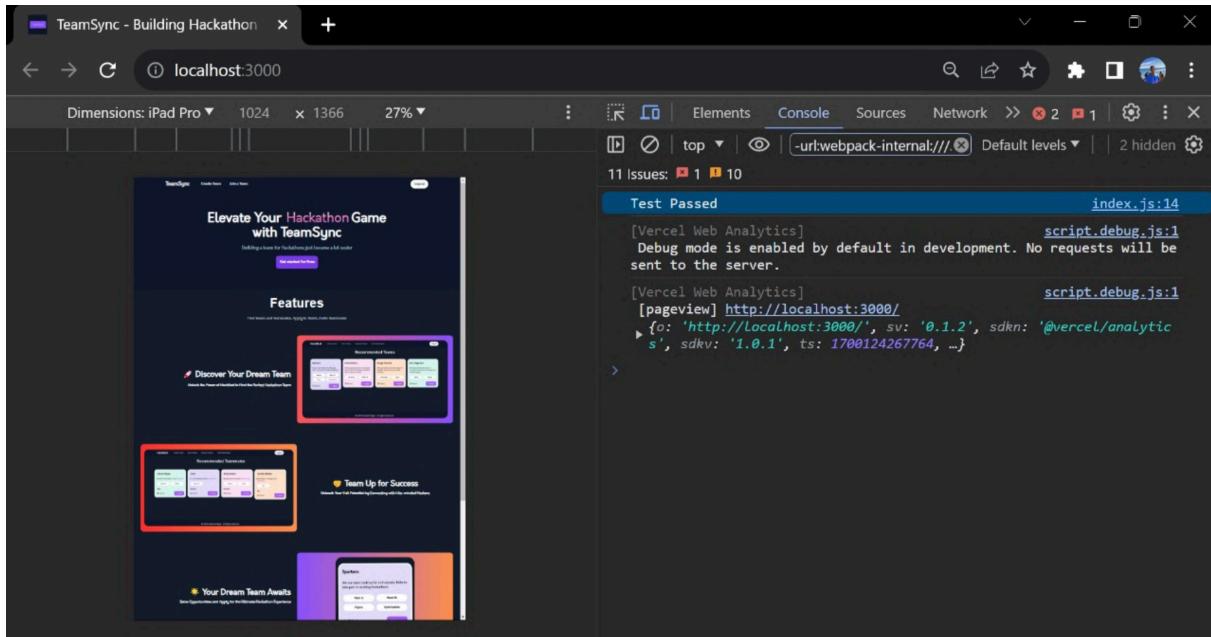
System Test Case ID	Requirement IDs	Test Case	Expected Output	Expected Results	Status
STC - 01	REQ-1, REQ-12	Load Dashboard After Login Responsively	Dashboard components load without errors on various screen sizes	Verify that user-specific information and key features are displayed correctly	Pass
STC - 02	REQ-2, REQ-3	Submit Form with Valid Input	Form submission success	Verify that the form is accepted and the user is redirected accordingly	Pass
STC - 03	REQ-4, REQ-8	Save Data to Database through Form Submission	Data saved correctly in the database	Verify that user input is stored accurately in the database	In Progress
STC - 04	REQ-7, REQ-10	Access Event Display	Events displayed based on user input	Verify that events are displayed correctly according to user preferences	Pass

ACCEPTANCE TESTING

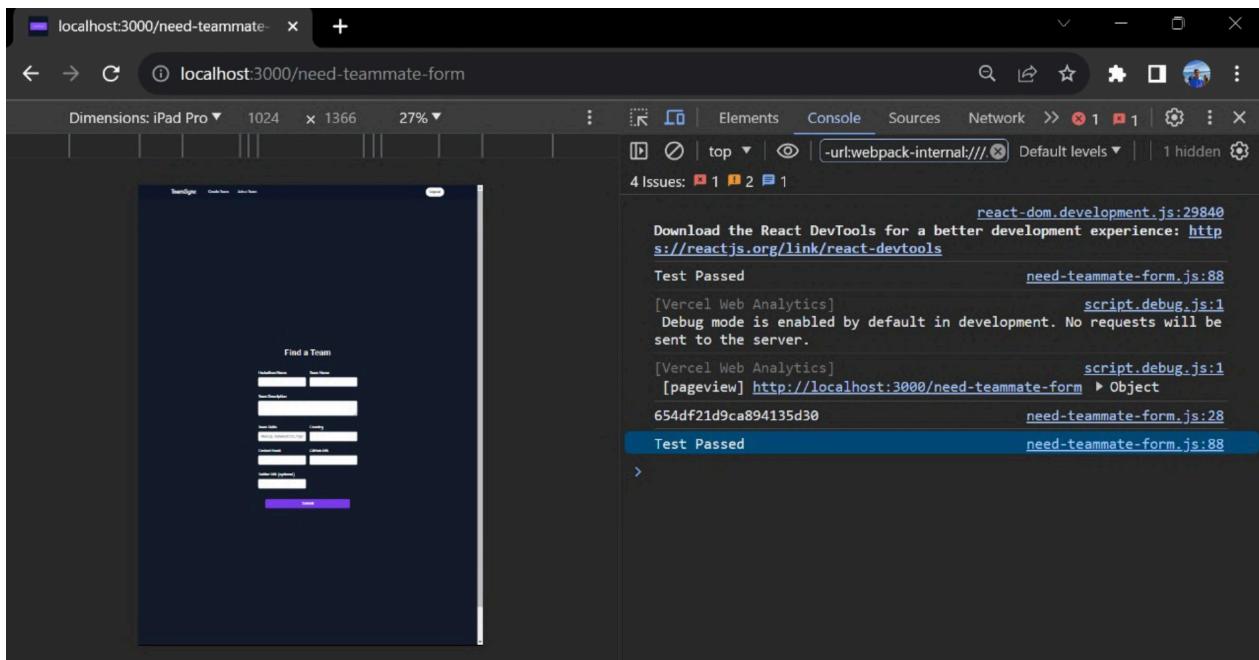
Acceptance Test Case ID	Requirement IDs	Test Case	Expected Output	Expected Results	Status
ATC - 01	REQ-1, REQ-12	User Accesses Landing Page Responsively	Landing page components load without errors on various screen sizes	Verify that images, styles, and links are displayed correctly	Pass
ATC - 02	REQ-2, REQ-3	User Logs in with Valid Credentials	Successful login and redirection	User is redirected to the main page after logging in	Pass
ATC - 03	REQ-4, REQ-8	User Submits Form and Verifies Data Storage	Data saved correctly in the database	Verify that user input is stored accurately in the database	In Progress
ATC - 04	REQ-7, REQ-10	User Accesses Event Display	Events displayed based on user input	Verify that events are displayed correctly according to user preferences	Pass

Testing Screenshots:

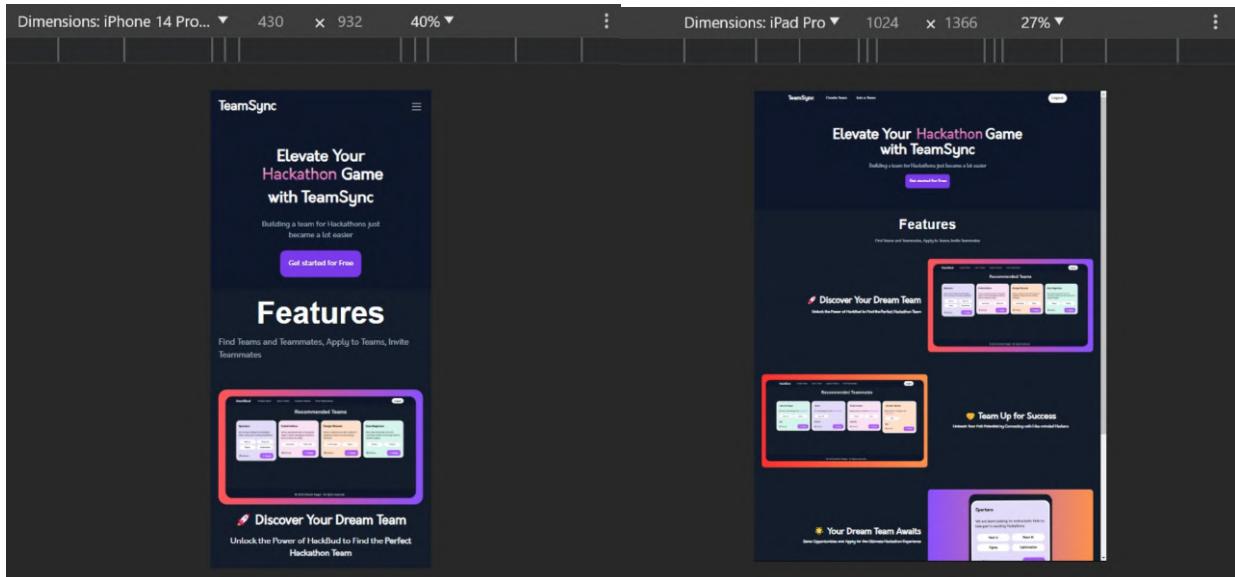
ATC-1 : Passed



ATC-4 : Passed



UTC-12 : Passed

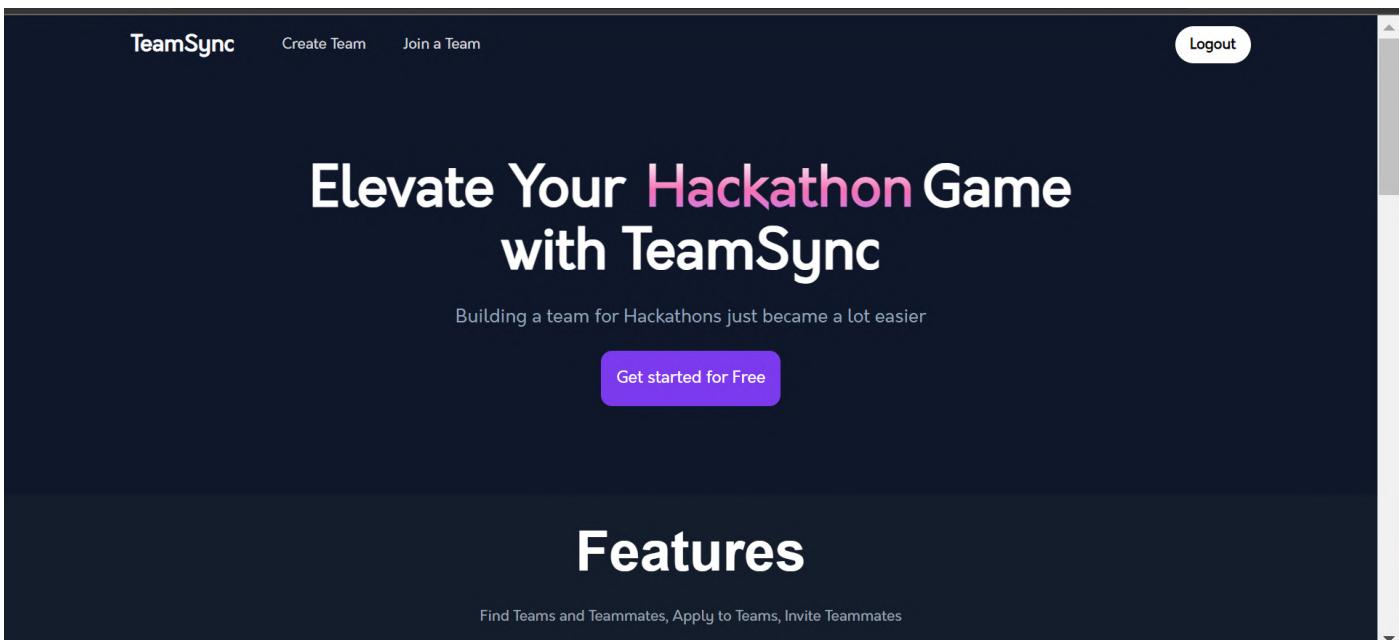


Screenshots of Output for TeamSync

PES University, RR Campus

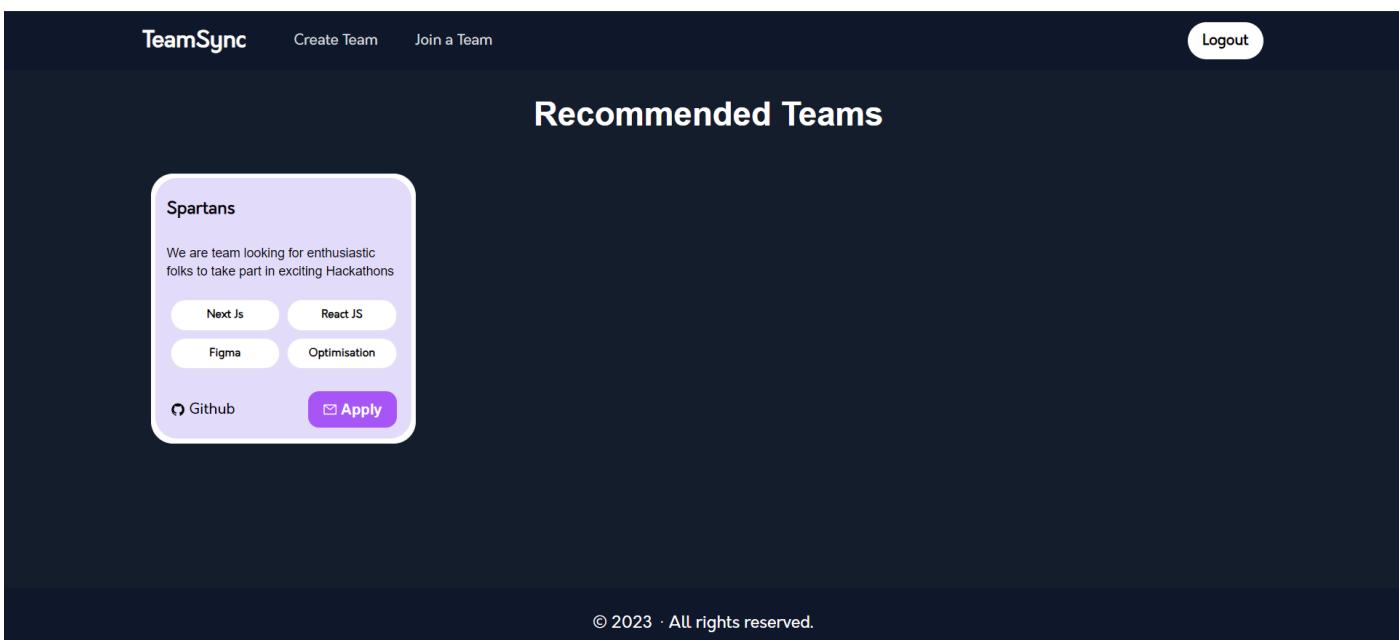
14 November 2023

Screenshot – 1 : Landing Page



The screenshot shows the homepage of the TeamSync platform. At the top, there is a navigation bar with the 'TeamSync' logo, 'Create Team', 'Join a Team', and a 'Logout' button. The main headline reads 'Elevate Your Hackathon Game with TeamSync' in large, bold, white font. Below it, a sub-headline says 'Building a team for Hackathons just became a lot easier'. A prominent purple button labeled 'Get started for Free' is centered. In the middle section, the word 'Features' is displayed in large white font. At the bottom, a sub-headline reads 'Find Teams and Teammates, Apply to Teams, Invite Teammates'.

Screenshot – 2 : Team Display Page



The screenshot shows a team profile page for 'Spartans'. The header includes the 'TeamSync' logo, 'Create Team', 'Join a Team', and a 'Logout' button. The main title is 'Recommended Teams'. The 'Spartans' team card features the team name at the top, followed by a description: 'We are team looking for enthusiastic folks to take part in exciting Hackathons'. Below the description are several skill tags: 'Next Js', 'React JS', 'Figma', and 'Optimisation'. At the bottom of the card are two buttons: a blue 'Github' button and a purple 'Apply' button. At the very bottom of the page, there is a copyright notice: '© 2023 · All rights reserved.'

Screenshot – 3 : Recommending Teammates.

The screenshot shows a dark-themed interface for 'TeamSync'. At the top, there are navigation links: 'Create Team' and 'Join a Team' on the left, and 'Logout' on the right. Below this, the title 'Recommended Teammates' is centered. The page displays five cards, each representing a recommended teammate:

- Ashish Pandare**: Full stack web developer. Skills: React, Tailwind, Mern, Javascript. Location: India. GitHub link. Invite button.
- Taranjot Singh**: I know it. Skills: React JS. Location: India. GitHub link. Invite button.
- Ananya**: 4th year undergrad from IIT Bh. Skills: NLP, ML, DeepLearning. Location: India. GitHub link. Invite button.
- Utkarsh**: Hey there I am looking for tea. Skills: React JS, Python. Location: India. GitHub link. Invite button.
- Aman Kumar**: xyuagoya. Skills: React. Location: India. GitHub link.
- Y**: y. Skills: Y. Location: Y. GitHub link.

Screenshot – 4 : Teammates Form.

The screenshot shows a dark-themed interface for 'TeamSync'. At the top, there are navigation links: 'Create Team' and 'Join a Team' on the left, and 'Logout' on the right. Below this, the title 'Find your Teammates' is centered. The form consists of several input fields:

- Full Name: Input field.
- Description: Input field.
- Skills: Input field containing 'Nextjs, TailwindCSS, Figma'.
- Country: Input field.
- Contact Email: Input field.
- GitHub URL: Input field.
- Submit: A large purple button at the bottom.

Screenshot – 5 : Team Form.

TeamSync Create Team Join a Team Logout

Find a Team

Hackathon Name Team Name

Team Description

Team Skills Country

Next.js, TailwindCSS, Figma

Contact Email GitHub URL

Twitter URL (optional)

Submit

Screenshot – 6 : Home Page (2).

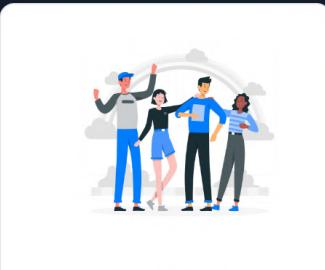
localhost:3000/choices +

localhost:3000/choices

TeamSync Create Team Join a Team Logout

Where Ideas Flourish Together

At the heart of progress lies collaboration. Join a community of driven individuals eager to unite their talents, share ideas, and redefine innovation. Together, we will pave the path to success.



I'm looking for a Team



I'm looking for Teammates

© 2023 All rights reserved

Screenshot – 7 : Home Page (3).

The screenshot displays the HackBud application interface on a dark-themed browser window. At the top, there's a navigation bar with links for 'Create Team', 'Join a Team', 'Explore Teams', and 'Find Teammates'. A user profile picture is visible in the top right corner.

Recommended Teams:

- Spartans**: We are a team looking for enthusiastic folks to take part in exciting Hackathons. Skills: Next.js, React.js, Python, Optimization. GitHub: [GitHub](#), Apply: [Apply](#).
- CodeCrafters**: We're a positive team of developers eager to tackle challenging hackathons and turn ideas into reality. Skills: Java Script, HTML, CSS. GitHub: [GitHub](#), Apply: [Apply](#).
- Design Wizards**: We're a creative bunch with a focus for designing intuitive and eye-catching interfaces. Skills: UI/UX Design, Python. GitHub: [GitHub](#), Apply: [Apply](#).
- Data Magicians**: We're data enthusiasts who love crunching numbers and turning data into valuable insights. Skills: Python, Pandas. GitHub: [GitHub](#), Apply: [Apply](#).

Recommended Teammates:

- Utkarsh Nagar**: Hey there! I am looking for team members. Skills: React.js, Python. Location: India. GitHub: [GitHub](#), Invite: [Invite](#).
- Anna**: Yo! I am looking for a team! Skills: React.js. Location: America. GitHub: [GitHub](#), Invite: [Invite](#).
- Emily Hacker**: Aspiring coder on a quest to code. Skills: Python, React. Location: Australia. GitHub: [GitHub](#), Invite: [Invite](#).
- Lily Byte Master**: Byte by byte, I'm coding my way to success. Skills: Java. Location: Italy. GitHub: [GitHub](#), Invite: [Invite](#).

Team Up for Success
Unleash Your Full Potential by Connecting with Like-minded Hackers