GDP – I SUMMER 2023

Professional Based Learning Explorer

**Team Members:**

Srinidhi Balmuri (Lead)

Samyuktha Pandula

Bala Harinadh Palavelli

Anil Kumar Palavelli

Divya sri Boppudi

Lella Pitchaiah

Krishna Sai Balupari

**Client Name**: Yu Zhao (Gary)

**Institution:** Northwest Missouri State University

1. **Table of Contents**.
2. **Introduction** (Anil Kumar Palavelli, Lella Pitchaiah)

The Northwest Professional-Based Learning (PBL) Project Explorer is an innovative platform designed to facilitate collaboration between faculties, students, and external stakeholders in developing and implementing PBL projects. This web-based application serves as a centralized hub where users can post their PBL project ideas, engage in discussions, and connect with potential team members.

The primary objective of the project explorer is to foster a dynamic and interactive environment that encourages the sharing of PBL project proposals and enables users to find suitable opportunities for participation. By creating a space for project posting, exploration, and collaboration, the platform aims to enhance the learning experience and promote real-world problem-solving skills.

* 1. **Purpose and scope:**

**Purpose:** The Northwest Professional-Based Learning (PBL) Project Explorer is a forum for faculty and students to publish and debate possible PBL projects. By letting people connect with project owners, take part in project discussions, and create teams to work on the projects, the initiative hopes to promote cooperation and involvement within the Northwest community.

**Scope:** The scope of the Northwest Professional-Based Learning (PBL) Project Explorer encompasses the development and implementation of a web-based application with the following key features:

**1. Project Posting:** Users, including faculties and students, can post their PBL project proposals on the platform. They provide project details such as title, description, objectives, timeline, and whether the project is public or private.

**2. Authentication:** Students can directly log in using their Student Identification (SID) credentials, while external users need to register and verify their email addresses to access the platform. Password reset functionality is also included.

**3. Project Approval:** Admin approval is required for each project to ensure its relevance and appropriateness before being made available on the platform.

**4. Project Categorization:** Projects can be categorized as public or private. Public projects are accessible to all users, including guests, while private projects are restricted to logged-in users only.

**5. Discussion Forum:** Each project has its own dedicated page with a discussion forum. Users can ask questions, make comments, and engage in discussions related to the project. Project owners and other users can respond to inquiries and participate in the conversation.

**6. Content Flagging:** Users have the ability to flag inappropriate comments or messages, which require admin review before being reinstated.

**7. User Profiles:** Users can create profiles where they can add their name, education details, email, and track the number of completed projects. Flagged users may have their accounts temporarily disabled until reviewed and reactivated by the admin.

**8. User Dashboard:** The user dashboard provides easy access to active and expired projects, as well as a listing of requests posted by the user and looking’s they have submitted. Pagination and filtering options enhance user experience and navigation within the platform.

**9. Requests and Looking’s:** Users can submit requests to work on other users' projects and create looking’s to find project opportunities for themselves. The platform allows users to view and manage their requests and looking’s.

**10. Project Request Management:** Project owners can view and manage the requests they receive for their projects, including reviewing request details, contacting potential team members through email or the internal messaging system, and making decisions on team formation.

The scope of the project includes the development of the above features, ensuring a user-friendly interface, robust security measures, and efficient project management processes. It also encompasses the necessary backend infrastructure, database management, and integration of email notifications for seamless communication.

* 1. **Overview, motivations, impact on the organization**

**Overview:** The Northwest Professional-Based Learning (PBL) Project Explorer is an online platform designed to facilitate the posting, exploration, and collaboration of PBL projects. It provides a centralized hub where faculties and students can share their project ideas, connect with potential team members, and engage in discussions. The platform offers features such as project posting, authentication, project categorization, discussion forums, user profiles, project and request listings, internal messaging system, and more.

**Motivations:**

**Enhancing Collaboration:** The project explorer aims to foster collaboration between faculties, students, and external stakeholders by providing a dedicated space for posting and exploring PBL projects. It encourages the sharing of ideas, expertise, and resources, leading to innovative and impactful project outcomes.

**Promoting Real-World Problem-Solving:** By engaging in PBL projects, faculties and students gain practical experience in tackling real-world problems. The platform motivates participants to apply their knowledge and skills to address complex challenges, bridging the gap between academic learning and real-world application.

**Impact on the Organization:**

**Increased Innovation and Creativity:** The project explorer encourages faculties and students to propose and engage in diverse PBL projects. This fosters a culture of innovation and creativity within the organization, leading to the development of novel solutions, ideas, and approaches.

**Enhanced Learning Outcomes:** The project explorer promotes active learning, critical thinking, problem-solving, and collaboration among faculties and students. By participating in PBL projects, they acquire practical skills, gain a deeper understanding of their field of study, and develop transferable competencies sought after by employers.

**Increased Visibility and Reputation:** The project explorer serves as a showcase platform for the organization's PBL initiatives. By highlighting the innovative projects and successful outcomes achieved through the platform, the organization can enhance its reputation, attract talented faculties and students, and demonstrate its commitment to experiential learning.

**Efficient Project Management:** The platform streamlines the process of project management, including project posting, request handling, and communication between project owners and potential team members. This efficiency improves project coordination and ensures a smooth workflow for all stakeholders involved.

In summary, the Northwest Professional-Based Learning Project Explorer has the potential to significantly impact the organization by promoting collaboration, fostering innovation, strengthening industry-academia ties, enhancing learning outcomes, and improving project management processes. By providing a platform for project exploration and sponsorship, the organization can cultivate a culture of experiential learning, practical application, and impactful problem-solving.

1. **Project Management Plan** (Bala Harinadh palavelli, Divya sri Boppudi)
   1. **Team organization:**

|  |  |
| --- | --- |
| ***Name*** | ***Role*** |
| Srinidhi Balmuri | Team Lead |
| Samyuktha Pandula | Full Stack Developer |
| Anilkumar Palavelli | Full Stack Developer |
| Krishna Sai Balupari | Back end |
| Bala Harinadh Palavelli | Tester |
| Lella Pitchaiah | Front end |
| Divya Sri Boppudi | Database |

* 1. **Risk Analysis:**

Performing a risk analysis for the "Project Explorer" involves identifying potential risks and assessing their likelihood and impact on the project's success. Here are some potential risks associated with the project:

**1. Security Risks:**

* **Data Breach:** There is a risk of unauthorized access to user data, such as personal information and project details, if proper security measures are not implemented.
* **User Authentication:** Weak authentication mechanisms could result in unauthorized access to user accounts or data.

**2. User Adoption and Engagement:**

* **Low User Engagement:** Users may not actively participate in discussions or post projects, resulting in a lack of engagement within the platform.
* **Limited User Base:** There is a risk that the platform may not attract a sufficient number of users, limiting the availability of projects and requests.

**3. Admin and Moderation Risks:**

* **Admin Approval Bottleneck:** If the admin approval process for posting projects is slow or inefficient, it may cause delays in project availability on the platform.
* **Inappropriate Content:** The discussion forum and messages may be prone to inappropriate or malicious content, requiring active moderation to maintain a positive user experience.

**4. Usability and User Experience:**

* **Complex User Interface:** A complicated or confusing user interface may hinder user adoption and result in poor user experience.
* **Performance Issues:** The platform may face performance issues if it cannot handle a large number of concurrent users or if there are inefficiencies in data retrieval and processing.

**5. External Dependencies:**

* **Third-Party Services:** Dependencies on external services, such as email verification or document storage, may introduce risks related to service availability, reliability, or compatibility.
  1. **Software and Hardware requirements:**

**Software Requirements:**

**Operating System:** The project should be compatible with common operating systems such as Windows, macOS, and Linux.

**Web Browsers:** The application should be compatible with popular web browsers like Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

**Development Tools:** Visual Studio Code, Eclipse, or JetBrains IntelliJ IDEA.

**Version Control System:** Git and platforms like GitHub or GitLab for collaborative code management.

**Communication and Collaboration:** Microsoft Teams or Zoom for team communication and collaboration.

**Database Management:** MySQL.

**Hardware Requirements:**

**Computer:** Each team member should have a computer system capable of running the required software tools and development environments efficiently.

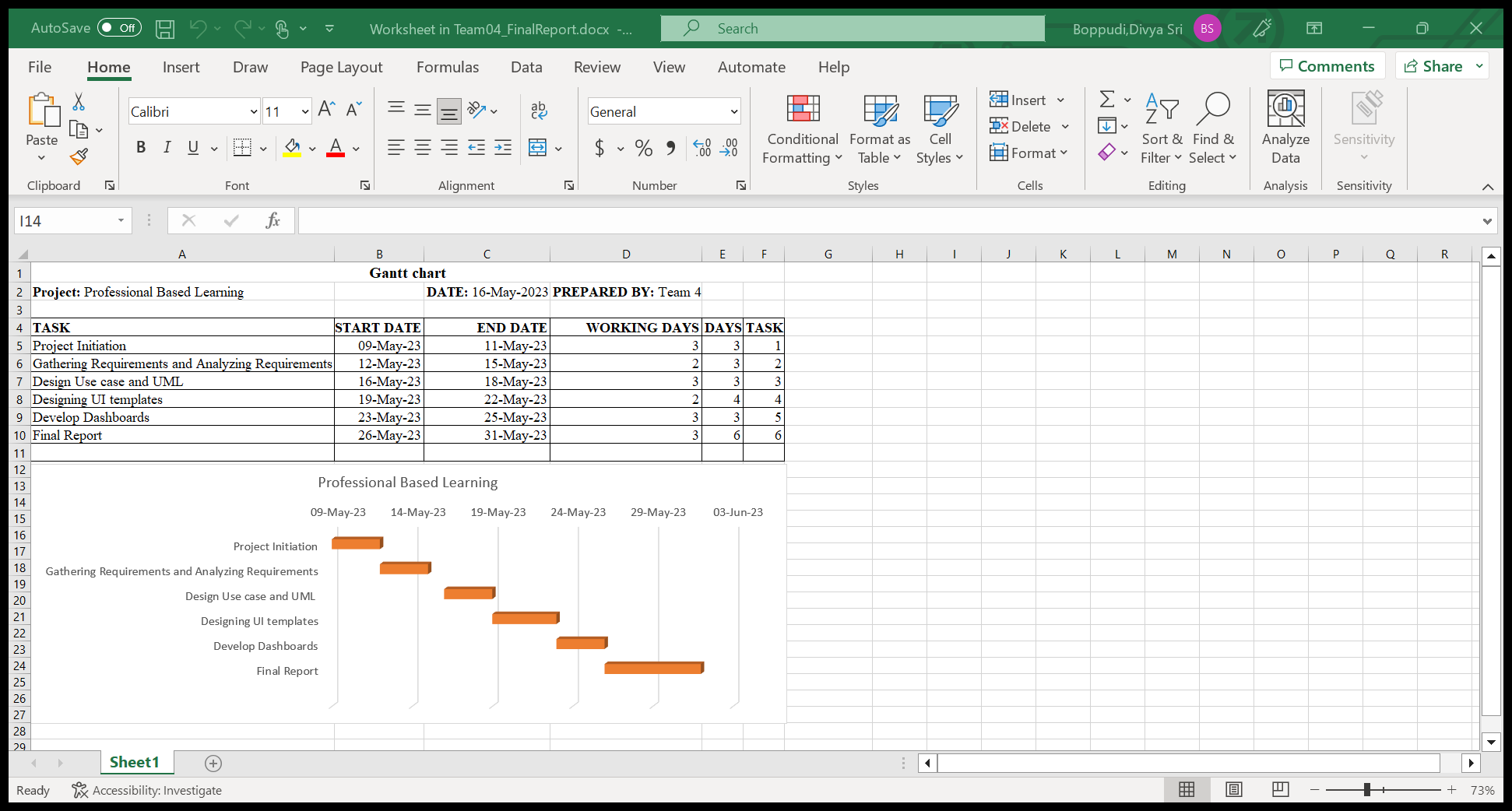
**Processor and Memory:** A modern processor with sufficient RAM (e.g., 8GB or more) to handle development tasks.

**Storage:** Adequate storage capacity to store the project files, development tools, and dependencies.

**Internet Connection:** A stable internet connection is necessary for accessing online resources, collaborating with team members, and testing the web application.

* 1. **Gantt Chart**

****

****

1. **Requirement Specification**(Srinidhi Balmuri, Krishna Sai Balupari )
   1. **Stakeholders of the System**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Name and Signature*** | ***Role*** | ***Internal/External*** | ***Position*** | ***Contact Information*** |
| Srinidhi Balmuri | Team Lead | Internal | Team Lead | [S555373@nwmissouri.edu](mailto:S555373@nwmissouri.edu) |
| Samyuktha Pandula | Team Member | Internal | Full Stack Developer | [S548204@nwmissouri.edu](mailto:S548204@nwmissouri.edu) |
| Bala Harinadh Palavelli | Team Member | Internal | Tester | [S549411@nwmissouri.edi](mailto:S549411@nwmissouri.edi) |
| Anil Kumar Palavelli | Team Member | Internal | Full Stack Developer | [S549406@nwmissouri.edu](mailto:S549406@nwmissouri.edu) |
| Divya Sri Boppudi | Team Member | Internal | Database | [S556453@nwmissouri.edu](mailto:S556453@nwmissouri.edu) |
| Lella Pitchaiah | Team Member | Internal | Frontend Developer | [S550608@nwmissouri.edu](mailto:S550608@nwmissouri.edu) |
| Krishna Sai Balupari | Team Member | Internal | Backend Developer | [S554235@nwmissouri.edu](mailto:S554235@nwmissouri.edu) |
| Gary Zhao | Client | External | Client | [ZHAO@nwmissouri.edu](https://nwmissouri-my.sharepoint.com/personal/s549411_nwmissouri_edu/Documents/ZHAO@nwmissouri.edu) |
| Dr. Aziz Fellah | Manager | Internal | Manager | [afellah@nwmissouri.edu](mailto:afellah@nwmissouri.edu) |

* 1. **Graphical UML Use Case Diagrams**

**Use Case Diagram:**

A picture containing text, diagram, line, screenshot

Description automatically generated

A diagram of a project

Description automatically generated with low confidence

A picture containing text, diagram, cartoon, line

Description automatically generated

**UML Diagram:**

A picture containing text, diagram, map, plan

Description automatically generated

* 1. **Nonfunctional Requirements**

Non-functional requirements specify the characteristics and qualities that the Northwest Professional-Based Learning (PBL) Project Explorer should possess. Here are some non-functional requirements for the project:

**Usability:** The platform should have an easy-to-use design that makes it simple for teachers, students, and other users to find their way around, post projects, join talks, and handle their accounts.

**Performance:** The system should be quick to respond and be able to handle a lot of users at once. It should have fast response times for sharing projects, reading, and other tasks to make sure users have a good time.

**Security:** To safeguard user data, prevent illegal access, and guarantee system integrity, the platform should have strong security mechanisms in place. This comprises safe systems for authentication and authorization, encryption of private information, and defense against widespread security flaws.

**Privacy:** The platform must respect user privacy and adhere to all applicable data protection laws. Users should have control over their data, including the opportunity to amend or remove it as required, and personal information should be managed securely.

**Accessibility:** The platform should follow accessibility guidelines to make sure that people with disabilities may use it successfully. This includes supporting keyboard navigation, offering alternate text for pictures, and being compatible with assistive technology.

1. **Architecture**(Samyuktha Pandula, Bala Harinadh Palavelli)
   1. **Architecture Model, Components and style(s) used.**

**Architecture Model:**

A suitable architecture model for "Project Explorer" could be a three-tier architecture, separating the application into three distinct layers: presentation layer, application logic layer, and data storage layer.

**Presentation Layer:**

This layer focuses on the user interface and handles the interaction between the users and the application. It includes components such as web pages, forms, navigation menus, and client-side scripts. The presentation layer communicates with the application logic layer to retrieve and display data.

**Application Logic Layer:**

This layer contains the business logic and processing components of the application. It handles the processing of user requests, validation, and business rules. It communicates with the presentation layer for data retrieval and updates.

**Data Storage Layer:**

This layer is responsible for managing and storing the application's data. It includes components such as a database or any other storage mechanism required for storing project details, user information, and other relevant data.

**Components:**

Based on the project requirements, here are some key components that can be part of the "Project Explorer" application:

**User Interface (Presentation Layer):**

Web pages and UI components for posting projects, viewing project details, and participating in the discussion forum. Forms for user registration, login, project posting, and request submission. User profile pages for managing personal information and project history.

**Application Logic Layer:**

Controllers or handlers to process user requests, validate inputs, and apply business rules. Authentication and authorization components to manage user access and permissions. Request handling components for managing project requests and approvals.

**Data Storage Layer:**

Database management system (e.g., MySQL) to store project details, user information, and related data. Database access components or Object-Relational Mapping (ORM) frameworks for interacting with the database.

**Architecture Style(s):**

For the "Project Explorer" project, the following architecture styles can be considered:

**Client-Server Architecture:**

The application can follow a client-server architecture where the client (web browser) communicates with the server to request and receive data.

The client-side code (HTML, CSS, JavaScript) interacts with the server-side components via API calls.

* 1. **Design: Algorithms used, interfaces used, rational for your design**

**Algorithms Used:**

**Project Filtering and Search:** The application can employ algorithms for filtering and searching projects based on various criteria such as project status (active, expired), project category, keywords, or user-specific preferences. These algorithms can efficiently retrieve and present relevant projects to users.

**Request Matching:** When a user posts a project, the application can use algorithms to match project requirements with the skills and interests of potential team members. This can involve analyzing user profiles, project details, and using techniques like keyword matching or machine learning-based recommendations.

**User Authentication and Authorization:** The application can use secure algorithms for user authentication, ensuring that only authorized users can access their accounts and perform actions within the system. This typically involves techniques like hashing and encryption to protect sensitive user information.

**Interfaces Used:**

**User Interface (UI):** The application's user interface is a critical interface for users to interact with the system. It should be designed to be intuitive, user-friendly, and provide a seamless experience for posting projects, submitting requests, and participating in the discussion forum.

**API Interfaces:** "Project Explorer" can provide well-defined and documented APIs to interact with the backend server. These APIs enable communication between the client-side (e.g., web browser) and server-side components, facilitating operations such as project posting, request submission, and retrieval of project details.

**Database Interfaces:** The application needs interfaces to interact with the database management system. These interfaces can be implemented using database query languages (e.g., SQL) or Object-Relational Mapping (ORM) frameworks to perform data storage, retrieval, and manipulation operations.

**Rationale for the Design:**

The design choices for the "Project Explorer" application are based on several factors:

**User Experience:** The user interface is designed to provide a user-friendly experience, making it easy for users to post projects, submit requests, and engage in discussions. Intuitive interfaces can encourage user engagement and facilitate efficient interaction with the application.

**Efficiency and Scalability:** Algorithms used for project filtering, request matching, and search functionality are designed to provide efficient and accurate results, ensuring that users can find relevant projects and suitable team members quickly. Scalable design considerations allow the application to handle increased user traffic and growing data volumes.

**Security:** The use of secure algorithms and interfaces for authentication and authorization helps protect user data, prevent unauthorized access, and ensure the privacy and security of user accounts.

**Integration:** The application can leverage APIs and interfaces to integrate with external services, such as email verification systems or storage services for file attachments. This enables seamless communication and enhances the functionality of the application.

**Maintainability and Extensibility:** The design aims to ensure that the application can be easily maintained and extended in the future. The use of well-defined interfaces and modular components allows for flexibility in making updates, adding new features, or integrating with additional systems.

1. **Technology Used**(Anilkumar Palavelli,Srinidhi Balmuri)
   1. **Front-End technologies:**

HTML (Hypertext Markup Language) will continue to be used for creating the structure and content of web pages. It provides the foundation for organizing and displaying the different elements of the PBL project explorer, such as project listings, user profiles, and project details.

CSS (Cascading Style Sheets) will be utilized to enhance the visual appearance and layout of the PBL project explorer. It will be responsible for styling the various HTML elements, including colors, fonts, spacing, and overall design. CSS ensures a consistent and appealing presentation of the web application.

JavaScript will be employed for adding interactivity and dynamic behavior to the PBL project explorer. It enables features such as form validation, real-time updates, and user interactions. JavaScript will play a crucial role in enhancing the user experience and making the web application more engaging and responsive.

React, a JavaScript library, will be integrated into the project to facilitate the development of the user interface. Reacts component-based architecture allows for the creation of reusable UI components, making the code more modular and maintainable. With React, developers can efficiently manage the state and rendering of the PBL project explorer, resulting in a smoother and more efficient user experience.

By combining HTML, CSS, JavaScript, and React, the Professional-Based Learning (PBL) project will benefit from a robust and dynamic web application. The integration of React will provide enhanced performance, code organization, and reusability, enabling the development team to build a scalable and interactive platform for faculty, students, and the CITE office to post, discuss, and implement PBL projects effectively.

* 1. **Back-End Technologies:**

**Java:** Java can be used to build the backend server-side components of the application. This includes handling requests, processing data, interacting with the database, and implementing business logic. Java's robustness, scalability, and extensive libraries make it a suitable choice for developing the core functionality of the application.

**Spring boot Framework:** Spring Boot provides a convenient and efficient way to develop RESTful APIs. You can define API endpoints using Spring MVC annotations, handle incoming requests, and return appropriate responses. Spring Boot's auto-configuration capabilities simplify the setup and configuration of the API layer.

* 1. **Database:**

MySQL : MySQL can store project details, user profiles, project requests, discussions, and other relevant data. It provides a reliable and scalable solution for persisting application data.

1. **Testing** (Divya sri Boppudi, Krishna sai Balupari)
   1. **Features to be tested:**

1) Post a new project:  
- Verify successful project submission when all required fields are filled correctly.  
- Test project submission with missing or invalid inputs, ensuring appropriate error messages are displayed.  
- Check if the project availability date is correctly recorded and visible on the website.  
- Test posting a project as both public and private simultaneously, ensuring it is not allowed.  
- Verify that the project status is set to "pending" until it receives admin approval.

 2) Authentication:  
- Test successful login with a valid Student ID (SID).  
- Verify successful registration process for outsiders, including receiving a verification email.  
- Test email verification process and ensure successful login after verification.  
- Verify the password reset functionality allows users to reset their password and log in with the new password.

3) Posted project full page with discussion forum:  
- Verify that guests and logged-in users can view the full page of a posted project, including details and the discussion forum.  
- Test user participation in the discussion forum by posting questions and comments.  
- Check if flagged comments require admin approval before being displayed again.  
- Test flagging messages in the discussion forum and ensure they are handled by the admin appropriately.  
- Verify that the "View Requests" button allows the project owner to see all the requests placed on the project.

4) User profile page:  
- Verify that the user's profile page displays all the major details, such as name, education details, email, and number of completed projects.  
- Test visibility of the profile page to all users and ensure flagged users' accounts are disabled until reactivated by the admin.  
- Check if the profile page is updated correctly when the user's information changes.

5) Projects page on user dashboard:  
- Verify that the user's dashboard displays active and expired projects separately.  
- Test filtering functionality to filter active projects based on project status (approved, pending, rejected).  
- Check if pagination is implemented correctly, allowing users to navigate through all the projects.  
- Verify that clicking on a project redirects the user to the page displaying the list of requests for that project.

6) Requests posted page on user dashboard:  
- Verify that all requests posted by the user for other projects are listed, including pending, approved, and rejected requests.  
- Test pagination functionality to ensure smooth navigation through the requests.

7) Looking’s page on user dashboard:  
- Verify that all looking’s posted by the user are listed.  
- Test pagination functionality to ensure smooth navigation through the looking’s.

8) Replies posted on user dashboard:  
- Verify that all replies posted by other users for the user's looking’s are listed.  
- Test pagination functionality to ensure smooth navigation through the replies.

9) Requests listing of a single project:  
- Verify that all requests received for a specific project are listed.  
- Test search functionality by person name to find specific requests.  
- Check if the username is a hyperlink that navigates to the user's profile page.

10) Request full page posted by the user:  
- Verify that the request full page displays all the details filled in the request submission form.  
- Test downloading of any attached documents.  
- Test the contact functionality through email or the internal message system.

* 1. **Criteria for passing and failing test.**

**1) Post a new project:**

**Passing Test Case:**

- User logs in successfully.

- User fills in all the required fields in the project submission form.

- User selects a specific date until which the project should be available.

- User chooses the project as public or private.

- Project is submitted successfully and awaits admin approval.

**Failing Test Case:**

- User tries to post a project without logging in.

- User leaves some required fields blank in the project submission form.

- User selects an invalid date for project availability.

- User tries to post a project as both public and private simultaneously.

**2) Authentication:**

**Passing Test Case:**

- Student logs in with their valid Student ID (SID) successfully.

- Outsider registers as a new user and receives a verification email.

- User verifies their email and successfully logs in.

- User resets their password and can log in with the new password.

**Failing Test Case:**

- Student enters an invalid Student ID during login.

- Outsider enters incorrect registration details and fails to receive a verification email.

- User enters an incorrect verification code during email verification.

- User enters an incorrect password during password reset.

**3) Posted project full page with discussion forum:**

**Passing Test Case:**

- Guest/user can view the project full page and read the project details.

- User logs in and participates in the discussion forum by posting questions and comments.

- User flags a comment and it requires admin approval to be visible again.

- User flags a message and it is handled by the admin.

- Project owner can view all requests placed on the project by clicking the "View Requests" button.

**Failing Test Case:**

- Guest/user cannot view the project full page and encounters an error.

- User cannot participate in the discussion forum without logging in.

- Flagged comments/messages do not require admin approval to be visible again.

- Project owner cannot view the requests placed on the project due to a system error.

**4) User profile page:**

**Passing Test Case:**

- User's profile page displays their name, education details, email, and number of completed projects.

- All users can view the profile page.

- Flagged users' accounts are disabled until admin activates them.

**Failing Test Case:**

- User's profile page does not display the required information.

- Only the user can view their profile page, and others encounter an error.

- Flagged users' accounts are not disabled or reactivated properly.

**5) Projects page on user dashboard:**

**Passing Test Case:**

- User's dashboard displays active and expired projects separately.

- User can filter active projects by project status.

- User can navigate through pagination to view all projects.

- Clicking on a project redirects to the page with the list of requests for that project.

**Failing Test Case:**

- User's dashboard does not display the active and expired projects separately.

- Filtering active projects by project status does not work correctly.

- Pagination does not function properly, preventing users from viewing all projects.

- Clicking on a project does not redirect to the correct page.

1. **Deliverables**(Samyuktha Pandula, Bala Harinadh Palavelli)
   1. **Project: Website, Git-hub Repository**

**Render link:** <https://projectbasedlearningexplorer.onrender.com/login>

**user login:** user **password:** user123

**admin login:** admin **password:** admin123

**Git-hub link:** <https://github.com/srinidhi1404/PBLProjectexplorerTeam04>

* 1. **How it works**

The Northwest Professional Based Learning (PBL) Project Explorer is an online platform designed to facilitate collaboration between students and faculty on PBL projects. It serves as a hub for posting, discussing, and implementing potential PBL projects. Here is a brief explanation of how it works:

1. **Post a New Project**: Any logged-in user can post a new project onto the platform, specifying a date until which the project should be available on the website. These posts undergo an approval process by an administrator before they are made live. A user also has the option to make the project either public or private.
2. **Authentication**: Students can login directly using their Student ID (SID), while external users need to register and verify their email before they can login. Users can also reset their passwords if needed.
3. **Posted Project Full Page with Discussion Forum**: Each posted project has its own dedicated page with a discussion forum. Here, users can view the project details and participate in discussions related to it. Any questions, comments, or flags can be addressed here. The person who posted the project or any other user can answer the questions.
4. **User Profile Page**: Here, users can add or update their personal and professional details such as name, education, email, and number of completed projects. These profiles are visible to everyone and can be flagged if any inappropriate activity is detected.
5. **Projects Page on User Dashboard**: This page classifies all projects into 'active' or 'expired'. Active projects are currently open for requests and discussions. Users can filter these projects by their status - approved, pending, or rejected. Expired projects, on the other hand, are completed and no longer receive requests or active discussions.
6. **Requests Posted Page on User Dashboard**: This page lists all the requests made by the user for other projects on the platform. The status of these requests (pending, approved, or rejected) can also be seen here.
7. **Looking’s Page on User Dashboard**: This page lists all the 'looking’s' posted by the user. 'looking’s' are essentially project needs or requirements that the user is looking for help with.
8. **Replies Posted on User Dashboard**: This page lists all the replies received by the user's posted 'looking’s'.
9. **Requests Listing of a Single Project**: Each project page includes a list of all requests received for that project. A search bar is also available to find requests by person name.
10. **Request Full Page Posted by the User**: This page provides all the details filled in the request submission form by the user. Any documents attached to the request can be downloaded by the user.

In summary, the NW PBL Project Explorer is a robust platform that enables the creation, discussion, and implementation of PBL projects. It facilitates efficient and effective communication and collaboration between users, making it an essential tool for professional-based learning environments.