CAMPUS AGILE **CMPSC 487W: INTRO TO SOFTWARE ENG** **SEP 10, 2023** **“Empowering university students with real-world project management experience through an agile task and collaboration platform”**	**Group Members** **Dharmik Patel** **Deep Patel** **Dravya Patel** **Dev Patel** **Darsh Patel** **Love Patel**

Team Info & Policies:

Our team and roles:

- *Deep Patel (Front-end Lead)*
- *Dravya & Darshit Patel (Front-end Developers)*
- *Dharmik Patel (Back-end Lead*
- *Dev & Love Patel (Back-end Developers)*

Git Repository:

- *We have set up a private GitHub repository for our project, accessible at GitHub Repository Link. The repository serves as our version control system for code collaboration and includes all project-related files.*

Communication and Tools:

- ***Communication Platform:*** *We utilize Discord for real-time discussions, team meetings, and weekly updates. Discord enables us to maintain effective communication and quick responses among team members.*
- ***Version Control:*** *GitHub is our chosen platform for hosting the project's Git repository. This facilitates code versioning, collaborative development, and code reviews.*
- ***Issue and Task Tracking:*** *Jira Software is our primary tool for tracking project-related issues, tasks, and milestones. Jira helps us organize and manage our project workflow efficiently.*

Policies:

- *We expect all team members to actively participate in Discord discussions and weekly meetings to ensure effective communication.*
- *Git push requests to the repository are reviewed and approved by team leads (Deep Patel for front-end and Dharmik Patel for back-end) to maintain code quality and consistency.*
- *Task assignments and updates are tracked using Jira Software, ensuring transparency and accountability in project progress.*

Product Description:

CampusAgile is a revolutionary platform dedicated to equipping university students with invaluable real-world project management experience. Our mission is to empower the next generation of professionals by providing them with the tools and knowledge needed to excel in an ever-evolving job market. With CampusAgile, students gain access to a collaborative environment that seamlessly integrates agile methodologies, enabling them to break down complex projects, develop adaptability, master time management, and prioritize tasks effectively.

Key Features:

- ***Agile Task Management:*** *CampusAgile introduces agile methodologies to students, teaching them how to dissect intricate projects into manageable components. This not only fosters project adaptability but also hones critical skills in time management and task prioritization.*

- ***Real-World Simulations:*** *Our platform offers a secure space where students can engage in real-life scenarios, allowing them to practice decision-making and tackle project challenges head-on. Through these simulations, students build confidence and acquire essential management skills that prepare them for the professional world.*
- ***Industry Integration:*** *CampusAgile collaborates with various organizations, bridging the gap between academic learning and industry requirements. By working on real projects sourced from industry partners, students not only enhance their networking opportunities but also increase their employability by gaining hands-on experience in their chosen field.*

Major Features (MVP):

- ***Task Assignment and Tracking:*** *Easily assign and monitor tasks, facilitating efficient project collaboration.*
- ***Interactive Agile Boards:*** *Visualize project progress through interactive agile boards, enhancing transparency.*
- ***Resource Management:*** *Optimize resource allocation and track utilization to ensure project efficiency.*
- ***Performance Analytics:*** *Access comprehensive analytics to evaluate individual and team performance, driving continuous improvement.*

Stretch Goals:

- ***Mobile Application:*** *Develop a mobile app for on-the-go access, providing convenience and flexibility.*
- ***Integration with Popular Tools:*** *Integrate CampusAgile with widely-used productivity and collaboration tools to enhance workflow.*

CampusAgile aims to be the premier platform for students seeking to bridge the gap between academia and industry, offering an immersive learning experience that prepares them for successful careers.

Use Cases (Functional Requirements):

Use Case 1: User Task Assignment

- ***Actors:*** *User (Student or Faculty), System*
- ***Triggers:*** *User initiates the task assignment process.*
- ***Preconditions:*** *User must be logged in.*
- ***Postconditions (success scenario):***
  - *User assigns tasks to team members successfully.*
  - *Task details and assignments are updated in the system.*
- ***List of Steps (success scenario):***
  - *User logs in.*
  - *User navigates to the project.*
  - *User selects the "Assign Tasks" option.*
  - *User specifies task details (title, description, priority, due date, assignee).*
  - *User submits the task assignment.*
- ***Extensions/variations of the success scenario:***
  - *If the task assignment fails, the user receives an error message and can retry.*
- ***Exceptions (failure conditions and scenarios):***
  - *If the user is not logged in, they are prompted to log in before proceeding.*

Use Case 2: User Collaboration on a Task

- ***Actors:*** *User (Student or Faculty), System*
- ***Triggers:*** *User selects a task to collaborate on.*
- ***Preconditions:*** *User must be part of the project and the task must exist.*
- ***Postconditions (success scenario):***
    - *User successfully collaborates on the task.*
    - *Task updates are reflected in real-time.*
- ***List of Steps (success scenario):***
    - *User logs in.*
    - *User selects the project.*
    - *User navigates to the task.*
    - *User views task details and discussions.*
    - *Users add comments, attach files, or update task status.*
- ***Extensions/variations of the success scenario:***
    - *Users can mention other team members for quick collaboration.*
- ***Exceptions (failure conditions and scenarios):***
    - *If the task does not exist or the user is not part of the project, they receive an error message.*

Non-functional Requirements:

- ***Usability:*** *The platform should have an intuitive user interface to ensure users can easily navigate, collaborate, and manage tasks without extensive training.*
- ***Security:*** *User data, project information, and communication should be encrypted to protect sensitive information. User authentication and authorization mechanisms must be robust.*
- ***Scalability:*** *The system should handle an increasing number of users, tasks, and projects without significant performance degradation.*

External Requirements:

- ***Robustness:*** *The platform should gracefully handle errors such as invalid user input, preventing system crashes or data corruption.*
- ***Installability:*** *CampusAgile should be accessible via a public URL for web-based access, and installation instructions should be provided for standalone applications.*
- ***Buildability:*** *All components, including clients and servers, should be buildable from source code. Comprehensive documentation should be available for developers to contribute and enhance the system.*
- ***Scope Alignment:*** *The scope of the project should align with the available team resources, ensuring that tasks and milestones are manageable within the team's capacity.*

Team process description:

Semester-Long Development Process:

Software Toolset: For our semester-long development process, we have carefully selected a set of software tools to facilitate efficient and effective development. These tools align with agile methodologies and provide the necessary support for our project:

1. *Front-end Development:*
    - *React.js: We will use React.js for building our front-end user interface due to its component-based architecture and high reusability.*
    - *TypeScript: TypeScript will be employed to add type safety to our JavaScript code, reducing errors and enhancing maintainability.*

    - *HTML/CSS: Standard HTML and CSS will be used for structuring and styling our web pages.*
    - *GitHub: We will utilize GitHub for version control, enabling seamless collaboration and code management.*
    - *Jira: Jira will serve as our task tracking tool to manage project tasks, issues, and sprints.*
    - *Discord: Discord will be our primary communication platform for team discussions and coordination.*
2. *Back-end Development:*
    - *Node.js: We have chosen Node.js for our back-end development, as it aligns well with JavaScript used in React, offering a unified development experience.*
    - *Database Management: Back-end developers will handle database management, selecting an appropriate database system based on project requirements.*
    - *Server Deployment: Deployment of the back-end server will be managed by the back-end lead, ensuring smooth operation.*

Team Member Roles:

- *Deep Patel (Front-end Lead): Deep will be responsible for leading the front-end development efforts. His role includes defining the front-end architecture, establishing a cohesive design and color palette, and assigning tasks to front-end developers. Deep's experience in front-end development makes him well-suited for this role.*
- *Dravya & Darshit Patel (Front-end Developers): Dravya and Darshit will work closely with the front-end lead to implement front-end components based on the provided design and requirements. Their responsibilities include coding, testing, and ensuring a seamless user experience.*
- *Dharmik Patel (Back-end Lead): Dharmik will take charge of the back-end development, overseeing the server-side architecture, database management, and server deployment. His experience in back-end development and NLP makes him the ideal candidate for this role.*
- *Dev & Love Patel (Back-end Developers): Dev and Love will collaborate with the back-end lead to develop server-side functionalities and integrate the database. They will work on implementing the logic necessary for the platform's core functionality.*

Schedule and Milestones:

Our development process is structured around key milestones to ensure progress throughout the semester:

1. *Project Initiation (Week 1-2):*
    - *Define project scope, goals, and requirements.*
    - *Set up the development environment.*
    - *Establish communication channels.*
    - *Create a project plan and assign roles.*
2. *MVP Development (Week 3-6):*
    - *Develop a Minimum Viable Product (MVP) with core features.*
    - *Conduct initial testing and debugging.*
    - *Prepare for internal testing and review.*
3. *Feature Additions (Week 7-10):*
    - *Add additional features and functionalities.*
    - *Conduct thorough testing of new features.*
    - *Refine user interface based on feedback.*
4. *Testing and Refinement (Week 11-14):*

- *Conduct comprehensive testing, including user acceptance testing.*
- *Address any identified issues and bugs.*
- *Optimize performance and security.*
- *Finalize documentation and user guides.*

Major Risks:

1. *Technical Challenges: Complex technical issues or limitations in the chosen technologies could impede progress. We will address this by maintaining open communication within the team and seeking external expertise if needed.*
2. *Scope Creep: Expanding project scope beyond the defined requirements can lead to delays. We will closely monitor project scope and prioritize features based on their impact and feasibility.*
3. *Resource Limitations: Limited availability of team members or access to necessary resources may affect project timelines. We will mitigate this by allocating tasks effectively and having contingency plans in place.*

External Feedback:

External feedback will be most valuable during the MVP development and testing phases. We plan to seek feedback from professors, industry experts, and potential end-users to refine our platform. This feedback will help us identify usability issues, performance bottlenecks, and areas for improvement before the final release. We will gather feedback through surveys, user testing sessions, and expert reviews, incorporating valuable insights into our development process.