

CSC648-04-F24-T4

Project Name:

On-Board

Project Description:

On-Board is a web-based platform designed to support college students, professionals, and individuals undergoing career transitions, regardless of age, in preparing for their career journey. The platform offers personalized features such as profile creation, job matching, and access to alumni networks and industry leaders based on career aspirations. With tools for tracking job applications, conducting skill assessments, and receiving tailored career recommendations, OnBoard provides a comprehensive solution for navigating the employment path efficiently. Additionally, the platform integrates real-time job alerts and mentorship opportunities to further enhance career readiness.

Team:

Krushna Thakkar – Team Lead

Chetas Parekh- Scrum Master

Khayal Dobaria- Front-End Developer

Jimmie Wu- Back-End Dev

Shrey Rakesh Kevadia- Git Master

Csc- Software Engineering

| Milestone | Front-End | Back-End | Scrum | Team Lead | Git |
|---|--|--|--|--|---|
| Milestone 0 16 th September | <ul style="list-style-type: none"> - Home Page layout - Navbar - Basic structure of Body - About Page | <ul style="list-style-type: none"> - Set up MongoDB - Define basic user model - Create API structure with Express.js | <ul style="list-style-type: none"> - Create and add tasks for Milestone 0 - Set daily standup meetings and backlog | <ul style="list-style-type: none"> - Review and assign tasks - Ensure project setup on Jira | <ul style="list-style-type: none"> - Review all pull requests - Ensure proper branching |
| Milestone 1 30 th September | <ul style="list-style-type: none"> - Implement Login/Sign-up pages - Front-End form validation - Profile setup page | <ul style="list-style-type: none"> - Develop API for user registration/login - Store user data in MongoDB - Set up authentication using JWT | <ul style="list-style-type: none"> - Add the new tasks for Login/Sign-up - Monitor progress in standups | <ul style="list-style-type: none"> - Oversee coordination between front-end and back-end teams - Review task progress | <ul style="list-style-type: none"> - Merge PRs for Login/Sign-up - Ensure CI/CD setup for deployment |
| Milestone 2 14 th October | <ul style="list-style-type: none"> - Implement Dark Mode toggle - Create dynamic job listing page - Filter options for job listings | <ul style="list-style-type: none"> - Develop API for job listing (pull from Indeed API) - Store and fetch job listings dynamically | <ul style="list-style-type: none"> - Create backlog for job listings - Review and track the dark mode implementation | <ul style="list-style-type: none"> - Ensure smooth API integration and test - Host team check-ins for any blockers | <ul style="list-style-type: none"> - Check and merge PR for Dark Mode and job listings - Coordinate job listing API integration |
| Milestone 3 30 th October | <ul style="list-style-type: none"> - Build and finalize Profile Section - Enable users to edit profile details | <ul style="list-style-type: none"> - Store user profile data in MongoDB - Handle API for updating profile details | <ul style="list-style-type: none"> - Add profile-related tasks to the backlog - Ensure profile section is complete on time | <ul style="list-style-type: none"> - Facilitate discussions on data storage for job matching | <ul style="list-style-type: none"> - Review and merge profile section - Ensure data flow between front-end and back-end |
| Milestone 5 16 th November | <ul style="list-style-type: none"> - Implement Job Search functionality - Job cards with details (title, location, skills) | <ul style="list-style-type: none"> - Develop job search API - Integrate job search filters (location, skills) | <ul style="list-style-type: none"> - Add tasks for job search feature - Monitor progress on search functionality | <ul style="list-style-type: none"> - Coordinate with back-end to ensure job search filters work | <ul style="list-style-type: none"> - Review and merge PR for job search - Test job search integration |
| Milestone 6 30 th November | <ul style="list-style-type: none"> - Implement Job Application tracking feature - UI for tracking applied jobs | <ul style="list-style-type: none"> - Implement API for job application tracking - Store applied jobs in MongoDB | <ul style="list-style-type: none"> - Add final tracking tasks to the backlog - Ensure the job application tracking is complete | <ul style="list-style-type: none"> - Ensure everything is ready for the final release - Oversee the testing of the entire system | <ul style="list-style-type: none"> - Merge and deploy final milestone - Ensure full system is working |

MileStone-0

Tech Stack:

Front-End: **React.js**

Back-End: **Express.js**

Cloud server: **AWS/ Git Pages**

Operating System and Version Number: **MacOS Sonoma , Version: 14.6.1**

Database and Version Number: **MongoDB, Version= 2.3.1**

Web Server: **Render**

Version Control: **Git**

Documentation: **Confluence**

Scrum Management: **Jira**

Communication: **Slack + Discord**

API Testing: **Postman**

Java Script Linter: **Prettier**

IDE: **Visual Studio Code**

Sample Page: <https://csc648-04-team04.onrender.com>

Team Lead:

Krushna Thakkar

As the Team Lead of the On-Board project, Krushna Thakkar is responsible for the overall success of the project, ensuring that the team works efficiently towards meeting the project goals, deadlines, and quality standards. Krushna oversees the technical and functional aspects of the platform, coordinates cross-functional collaboration, and acts as a liaison between stakeholders and the development team.

Responsibilities:

- **Project Vision & Strategy:**
 - **Defining the Project Scope:** Works closely with stakeholders, including professors, industry advisors, and end users, to clearly define the scope, objectives, and key deliverables of the OnBoard platform.
 - **Setting Milestones:** Breaks down the project into key milestones, ensuring that each sprint has clear objectives and that the team stays aligned with the long-term project vision.
 - **Aligning the Team to Project Goals:** Communicates the project's goals and ensures that each team member understands their role in achieving these goals. Encourages collaboration and promotes the importance of quality in every task.

- **Leadership & Team Coordination:**
 - **Team Collaboration:** Fosters a collaborative and supportive environment within the team, ensuring that team members are motivated, understand their tasks, and work cohesively.
 - **Role Assignment:** Identifies each team member's strengths and assigns them tasks accordingly, ensuring efficient use of resources and skills.
 - **Conflict Resolution:** Addresses any issues or conflicts within the team, promoting constructive problem-solving and maintaining a positive team dynamic.

- **Decision-Making & Problem Solving:**
 - **Critical Decisions:** Leads the decision-making process on key technical and strategic choices such as tech stack selection, design patterns, or integration of third-party tools and APIs.
 - **Problem-Solving:** Provides guidance in resolving complex technical challenges faced by the team, helping to unblock issues in both front-end and back-end development.

- **Quality Assurance & Code Reviews:**
 - **Maintaining Code Quality:** Regularly reviews pull requests, ensuring the quality and consistency of the codebase. Enforces coding standards, best practices, and project-specific guidelines to ensure scalable and maintainable code.
 - **Technical Oversight:** Provides technical oversight across the entire stack (React.js for front-end, Express.js for back-end, MongoDB for database), ensuring that both front-end and back-end teams follow architectural and design principles.
- **Stakeholder Management:**
 - **Stakeholder Communication:** Acts as the primary point of contact for stakeholders, including professors and users, providing regular updates on project progress, roadblocks, and upcoming deliverables.
 - **Gathering Requirements:** Regularly meets with stakeholders to gather feedback, ensure alignment on requirements, and refine project priorities based on user needs.
 - **Demonstrating Progress:** Organizes regular demo sessions at key milestones to showcase the platform's progress and gather critical feedback for future development.
- **Risk Management:**
 - **Identifying Risks:** Proactively identifies potential risks, such as delays, scope creep, or technical hurdles, and formulates contingency plans to mitigate them.
 - **Timeline Management:** Monitors the project timeline and ensures that milestones are met, making adjustments to priorities and resources when necessary to avoid delays.
- **Mentorship & Growth:**
 - **Supporting Team Growth:** Acts as a mentor to team members, providing guidance on both technical and non-technical skills. Encourages learning and ensures that the team stays up-to-date with the latest technologies and best practices.
 - **Knowledge Sharing:** Promotes a culture of knowledge sharing within the team by organizing regular code reviews, technical discussions, and problem-solving sessions.
- **Cross-Functional Collaboration:**
 - **Coordinating Across Teams:** Ensures smooth communication and collaboration between the front-end, back-end, and DevOps teams. Works closely with the Git

Master and Scrum Master to ensure that everyone is aligned on project priorities and timelines.

- **Integration of New Features:** Coordinates the integration of new features and ensures that they are seamlessly implemented in the platform without disrupting existing functionalities.

- **Project Documentation & Reporting:**
 - **Maintaining Documentation:** Ensures that all aspects of the project, including architecture decisions, codebase changes, and deployment processes, are well-documented. This helps maintain project continuity and serves as a reference for future developers.
 - **Progress Reporting:** Provides regular reports on project progress, risks, and achievements to stakeholders and faculty, ensuring that everyone is kept informed about the project's status.

Front-End

Front-End Developer: Khayal Dobaria

The front-end of the OnBoard platform is designed to provide a seamless user experience, focusing on responsiveness, accessibility, and dynamic content presentation. The front-end team ensures that all visual elements are intuitive and perform smoothly across different devices and browsers.

Functionalities:

User Authentication & Profile Setup:

Login/Sign-Up: Implements secure authentication mechanisms, allowing users to create accounts using their email or OAuth (Google, LinkedIn) and securely log in.

Profile Creation: Develops detailed forms for students and professionals to create and edit their profiles. The profile includes:

- Personal details (name, email)
- Educational background (schools, degrees, majors)
- Work experience (roles, companies)
- Project experience (key projects, outcomes)
- Career goals and aspirations (desired roles, industries)
- Social links (LinkedIn, GitHub, personal websites)

Profile Picture Upload: Allows users to upload their profile pictures with image validation (file type, size limit) and cropping functionality.

- **Front-End Validation:** Ensures form fields are properly validated before submission, checking for required fields, email format, and data constraints, and providing real-time error feedback to the user.

Job Matching and Listing:

Job Search Interface:

- Presents dynamically fetched job listings tailored to user profiles using APIs like Indeed, LinkedIn, and Glassdoor.

- Each job card displays key details such as job title, company, location, required skills, and experience level.
- Allows users to filter and sort jobs by multiple criteria like location, job type, salary, and required experience.

Job Details Page:

- Shows in-depth job information, including job description, required qualifications, and an easy-to-navigate apply button that redirects users to external job applications.
- Option to save jobs to a personal "Saved Jobs" section for later viewing.
- **User Interface (UI) Design & Responsiveness:**
- Ensures a clean, user-friendly design that is responsive across various screen sizes (mobile, tablet, desktop) using a mobile-first approach.
- Implements a consistent color palette and typography aligned with the brand identity.

Performance Optimization:

- Uses lazy loading techniques to improve performance by loading images and job listings only when they are needed.
- Utilizes React.js's state management and hooks to efficiently manage and update the UI based on user interaction.

Cross-Browser Compatibility:

- Ensures the platform functions smoothly across different web browsers, including Chrome, Safari, Firefox, and Edge.
- **Accessibility (a11y):**
- Integrates accessibility best practices (ARIA labels, keyboard navigation, screen reader compatibility) to ensure the platform is usable by people with disabilities

Git Master

Shrey Rakesh Kevadia

The Git Master ensures the seamless collaboration of the team by maintaining a clean, organized Git repository and enforcing version control best practices.

Responsibilities:

Repository Management:

- Manages the Git repository hosted on GitHub, setting up the project's branch structure (feature branches, development, main) to ensure smooth collaboration and CI/CD workflows.
- Monitors pull requests and reviews code to ensure that only high-quality, tested code is merged into the main branch.

Git Workflows & Best Practices:

- Implements the GitFlow branching model, ensuring that team members follow standardized workflows for branching, committing, and merging.
- Facilitates regular code reviews and peer feedback before merges.

Conflict Resolution:

- Manages merge conflicts by identifying issues, communicating with the team, and assisting in the resolution of conflicting changes.

Continuous Integration (CI) Setup:

- Configures automated build and test pipelines using GitHub Actions, ensuring that code pushed to the repository is automatically tested and verified.
- Enforces pre-commit hooks using tools like Prettier and ESLint to standardize code format and style across the team.

Version Control Documentation:

- Documents Git workflows, commit message guidelines, and branching strategies for all team members.
- Tracks versioning of key platform milestones and releases

Scrum Master

Chetas Parekh

As Scrum Master, Chetas is responsible for facilitating Agile processes, ensuring smooth sprint execution, and promoting collaboration and efficiency within the team.

Responsibilities:

Sprint Planning & Facilitation:

- Leads sprint planning meetings, where the team discusses the upcoming sprint's goals, tasks, and deliverables. Ensures the team is clear on priorities and deadlines.
- Helps in breaking down large tasks into smaller, manageable stories or tasks and assigns them to team members according to their roles and expertise.

Daily Standups:

- Conducts daily standup meetings where each team member provides updates on their progress, challenges they're facing, and plans for the day. Ensures that blockers are discussed and addressed quickly.

Sprint Review & Retrospective:

- Organizes sprint reviews to showcase the completed work to stakeholders, gather feedback, and ensure that the platform is meeting user and business requirements.
- Facilitates sprint retrospectives to review what went well, what didn't, and how the team can improve in the next sprint.

Backlog Management:

- Works closely with the Product Owner to ensure the product backlog is prioritized based on business goals and stakeholder feedback. Ensures that the team is working on the highest-value features first.

Agile Coaching:

- Coaches the team on Agile and Scrum principles, ensuring they understand the importance of iterative development, continuous improvement, and collaboration.
- Monitors the team's velocity and suggests improvements to the development process to enhance productivity and reduce bottlenecks.

Risk & Issue Management:

- Proactively identifies and manages risks, helping the team anticipate potential challenges and come up with mitigation strategies.
- Ensures that any issues that arise during sprints are addressed promptly without affecting the sprint timeline.

Tool Management:

- Configures and manages Jira for task tracking, sprint planning, and reporting. Ensures that tasks are well-defined, estimated, and updated regularly by team members.
- Facilitates communication across different platforms, including Slack and Discord, to ensure team members stay connected and informed.

Back-End

Jimmie Wu

As the Back-End Lead for the OnBoard project, Jimmie Wu is responsible for designing, implementing, and maintaining the server-side components of the web application. This includes API development, database management, user authentication, and the integration of third-party services. Jimmie ensures that the back-end architecture is efficient, scalable, secure, and able to support all front-end interactions.

Responsibilities:

- **API Development & Route Handling:**
 - **RESTful API Design:** Designs and develops RESTful APIs to handle communication between the front-end and back-end systems. Defines clear and well-documented API endpoints that allow efficient interactions with the database and external services.
 - **Routing & Endpoints:** Handles all route management, creating endpoints for key functionalities such as user registration, login, job listing retrieval, job application tracking, and profile updates. Ensures that these routes are logically organized and adhere to RESTful principles.
 - **Error Handling:** Implements comprehensive error handling for the APIs, ensuring that proper HTTP status codes (e.g., 200, 404, 500) are returned and that the front-end receives clear and actionable error messages when issues arise.
- **Database Interaction & Management:**
 - **MongoDB Integration:** Designs and manages the database schema in MongoDB. Defines models for user profiles, job listings, and application data, ensuring data is stored efficiently and can be retrieved quickly.
 - **CRUD Operations:** Implements robust create, read, update, and delete (CRUD) operations for all data stored in MongoDB. This includes handling large amounts of data from users, jobs, and third-party APIs (like Indeed), ensuring it can be managed in real-time.
 - **Data Optimization:** Optimizes database queries to ensure fast and efficient data retrieval, especially for complex queries like filtering job listings based on location, skills, and experience.
- **User Authentication & Authorization:**
 - **JWT (JSON Web Token) Implementation:** Implements JWT for secure user authentication, ensuring that users can safely log in and maintain their sessions while interacting with the platform. Ensures that tokens are securely generated and validated.

- **Session Management:** Manages user sessions and ensures that users' access tokens remain valid, providing seamless user experiences without frequent reauthentication. Securely handles token refresh logic for long-running sessions.
- **Access Control:** Enforces role-based access control (RBAC), ensuring that different users (e.g., students, alumni, admins) have appropriate access levels to different resources or functionalities based on their permissions.
- **Business Logic & Data Validation:**
 - **Custom Business Logic:** Implements the core business logic that defines how data is processed and interacts with the database. This includes algorithms for job matching, sorting, and recommendations based on user profiles, skills, and career goals.
 - **Data Validation:** Validates incoming data from the front-end (e.g., user registration, job applications) to ensure that it meets the required formats and constraints. This prevents invalid data from being processed or stored in the database, ensuring data integrity.
- **Third-Party API Integration:**
 - **Job Listing APIs (Indeed):** Integrates third-party APIs such as Indeed to fetch real-time job listings and display them on the platform. This requires handling API requests, parsing the returned data, and transforming it into a format suitable for OnBoard's front-end.
 - **Real-Time Job Alerts:** Builds backend logic for real-time job alerts, ensuring users can subscribe to job alerts based on their profile data and receive notifications when matching jobs are posted.
 - **Mentorship & Alumni Network:** Integrates APIs to connect users with alumni and industry mentors. Handles data parsing and ensures secure interaction between OnBoard and these external platforms.
- **Data Security & Compliance:**
 - **Data Encryption:** Implements encryption for sensitive data such as passwords and personal user information stored in the database. Ensures secure transmission of data between the front-end and back-end using HTTPS.
 - **Security Best Practices:** Applies industry best practices for back-end security, including protecting against SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). Regularly audits code for security vulnerabilities.
 - **Compliance with Standards:** Ensures the application adheres to data privacy regulations like GDPR or CCPA, ensuring user data is handled with care and that users can manage their data (e.g., right to access or delete their data).

- **Scalability & Performance Optimization:**
 - **Performance Tuning:** Continuously optimizes the server-side code and database queries to improve response times, particularly for resource-heavy operations like job searching or profile updates.
 - **Load Balancing & Caching:** Implements strategies like load balancing to distribute server load and caching to improve the performance of frequently accessed data (e.g., job listings or user profiles).
 - **Scalable Architecture:** Designs a back-end architecture that can scale as the platform grows in terms of users and data, ensuring smooth performance even as the number of users or job listings increases significantly.
- **Testing & Quality Assurance:**
 - **Unit & Integration Testing:** Writes unit tests to validate individual components of the back-end and integration tests to ensure smooth interaction between the front-end and back-end. Uses testing tools like Mocha or Chai to automate tests and catch issues early in the development process.
 - **API Testing:** Conducts API testing using tools like Postman to ensure that all endpoints function as expected, handle edge cases correctly, and return appropriate error codes and messages.
- **Collaboration & Documentation:**
 - **Cross-Functional Coordination:** Works closely with the Front-End Developer (Khayal Dobarra), Git Master (Shrey Rakesh Kevadia), and Scrum Master (Chetas Parekh) to ensure that the front-end and back-end systems are properly integrated and function as a cohesive whole.
 - **Technical Documentation:** Ensures that all API endpoints, database models, and business logic are thoroughly documented for future developers and the team. This documentation provides a clear understanding of how the back-end is structured and how it should be maintained and extended.
 - **Version Control & CI/CD:** Collaborates with the Git Master to ensure that back-end changes are properly version-controlled, with meaningful commits and pull requests. Follows best practices for continuous integration and deployment (CI/CD) to deploy updates to the production environment smoothly.