**Week 1: Project Setup and User Authentication**

* **Objective**:
  + Set up the foundational structure of the project and implement user authentication with role-based access control (RBAC).
* **Tasks**:
  + Initialize Node.js project with Express, TypeScript, and MongoDB.
  + Implement user registration and login functionality with email and password.
  + Integrate JWT for secure authentication.
  + Define roles (e.g., Student, Supervisor) and implement RBAC.

**Week 2: Document Creation, Editing, and Document Upload**

* **Objective**:
  + Enable users to create, format, save, and upload documents in various formats.
* **Tasks**:
  + Integrate Quill as the rich-text editor for document creation.
  + Enable basic formatting options (e.g., bold, italics, underline, lists).
  + Implement backend API for saving and updating documents in the database.
  + Allow users to upload documents in .docx and .pdf formats.

**Week 3: Document Versioning and Collaboration (Part 1)**

* **Objective**:
  + Implement version control for documents and lay the groundwork for real-time collaboration.
* **Tasks**:
  + Set up versioning system in backend and create database schema for document versions.
  + Implement logic for creating new versions upon document save and store version metadata.
  + Integrate versioning with frontend UI and display version history.
  + Allow users to view version details and restore previous versions.

**Week 4: Document Versioning and Collaboration (Part 2)**

* **Objective**:
  + Finalize document versioning and establish real-time collaboration features.
* **Tasks**:
  + Ensure document versioning is fully functional and thoroughly tested.
  + Set up WebSockets for real-time collaboration between multiple users.
  + Implement basic permissions and roles (e.g., Student, Supervisor) for user access control.

**Week 5: Collaboration (Part 2)**

* **Objective**:
  + Enhance collaboration features, ensuring smooth real-time interactions.
* **Tasks**:
  + Test real-time collaboration with multiple users, different document types, and address any synchronization issues.
  + Implement additional collaboration features (e.g., chat, comments) to enhance user experience.
  + Conduct load testing to ensure the system handles a large number of simultaneous users.

**Week 6: Commenting and User Profiles**

* **Objective**:
  + Implement commenting functionality and create user profiles.
* **Tasks**:
  + Allow users to add comments to specific sections of a document and store comments in the database.
  + Create a user profile page for users to view and edit their profile details.

**Week 7: UI Styling and Finalization**

* **Objective**:
  + Enhance the visual appeal of the application and finalize features.
* **Tasks**:
  + Apply CSS and styles to improve the overall look and feel of the application.
  + Refine UI styling based on user feedback and usability testing.
  + Implement responsive design for different screen sizes.

**Week 8: Testing, Deployment, and Final Checks**

* **Objective**:
  + Ensure the application is thoroughly tested, deployed, and ready for use.
* **Tasks**:
  + Implement input validation and error handling for user interactions.
  + Choose a cloud platform (e.g., AWS, Heroku) for deployment and set up necessary configurations.
  + Deploy the application and perform final checks to ensure it meets performance and stability requirements.