OPT Portal Project Week 3 Deliverable:

Week 3 (August 26 - August 30)
Stage 3: Core Development and Testing
Objective: Develop core functionalities, integrate the frontend with the backend, and perform thorough testing, including the volunteer application process.

Backend Team

- **August 26 (Monday): Database Integration**
  - **Task:** Connect the application with MongoDB.
    - Ensure the MongoDB database is connected correctly with the backend server.
    - Verify that the schema design aligns with the collections and relationships established in previous weeks, including a new `Volunteers` collection.
    - Optimize database queries for efficiency and security, particularly focusing on reducing latency and ensuring data integrity.
    - Implement indexing strategies for frequently accessed collections (e.g., Users, Documents, Volunteers) to enhance query performance.

- **August 27 (Tuesday): User Authentication and Volunteer Registration**
  - **Task:** Develop API endpoints for user authentication (registration, login) and volunteer registration.
    - Implement the registration and login APIs with secure handling of user credentials (e.g., hashing passwords using bcrypt).
    - Add new API endpoints for volunteers to apply, capturing necessary details (e.g., personal information, availability).
    - Ensure that the registration process includes validation for email uniqueness and strong password policies.
    - Integrate JWT (JSON Web Token) for session management, including token issuance, expiration, and refresh strategies.
    - Conduct unit testing for each authentication and volunteer registration endpoint to ensure proper functionality and security.

- **August 28 (Wednesday): Authentication and Volunteer Application Flow**
  - **Task:** Implement the Login component with authentication flow and volunteer application process.
    - Develop the backend logic to support the authentication flow, including session handling and user role validation.
    - Create backend logic for volunteer application submission and approval processes.

- ■ Test the authentication and volunteer application processes thoroughly to ensure they handle various scenarios (e.g., invalid credentials, expired tokens, incomplete applications).
- ■ Implement error logging and monitoring for authentication and application submission failures to aid in troubleshooting.

- **August 29 (Thursday): Secure API Endpoints**
  - **Task:** Ensure API endpoints are secure and handle edge cases.
    - ■ Apply authentication middleware to protect sensitive endpoints, ensuring only authorized users and volunteers can access them.
    - ■ Implement role-based access control (RBAC) to restrict actions based on user roles (Admin, Student, OPT Coordinator, Volunteer).
    - ■ Test the middleware to ensure it handles edge cases, such as invalid tokens or unauthorized access attempts.
    - ■ Perform security testing using tools like OWASP ZAP to identify and mitigate vulnerabilities.

- **August 30 (Friday): Notification and Volunteer Management System Implementation**
  - **Task:** Implement Notification component for updates and alerts, including volunteer application status.
    - ■ Develop backend logic for sending notifications (e.g., volunteer application status updates, new opportunities).
    - ■ Integrate with a notification service (e.g., Firebase Cloud Messaging) to handle real-time alerts.
    - ■ Test the notification functionality across different user roles, ensuring that the correct messages are delivered to the intended recipients.

Frontend Team

- **August 26 (Monday): CRUD Operations Testing**
  - **Task:** Test basic CRUD operations for Users, Documents, and Volunteers collections.
    - ■ Verify that the frontend can successfully perform Create, Read, Update, and Delete operations on the Users, Documents, and Volunteers collections.
    - ■ Ensure that the data flow between the frontend and backend APIs is smooth and consistent.
    - ■ Conduct UI testing to validate that the CRUD operations are reflected correctly in the user interface.

- **August 27 (Tuesday): API Integration for Document Management and Volunteer Application**

- ○ **Task:** Develop frontend API integration for document submission, management, and volunteer application.
    - Implement the frontend logic for submitting documents and volunteer applications through forms, including validation for required fields.
    - Handle form submission errors gracefully, providing user feedback for common issues (e.g., file size limits, unsupported formats, incomplete volunteer applications).
    - Test the document management and volunteer application functionalities end-to-end, ensuring that submissions are reflected in the database.

- ● **August 28 (Wednesday): Dashboard and Volunteer Application Components Implementation**
    - ○ **Task:** Implement the Dashboard component to display the document list and volunteer application status.
        - Develop the frontend Dashboard component to fetch and display a list of documents submitted by the user and volunteer application statuses.
        - Ensure that the Dashboard reflects real-time data from the backend, with support for filtering and sorting options.
        - Test the Dashboard for usability, making sure that it is intuitive and user-friendly across all supported devices.

- ● **August 29 (Thursday): UI Testing, Volunteer Application, and Bug Fixes**
    - ○ **Task:** Test basic CRUD operations for Users, Documents, and Volunteers collections through the UI.
        - Perform thorough testing of the UI to identify and fix any bugs related to CRUD operations and volunteer application processes.
        - Validate that all user actions in the UI (e.g., submitting a form, updating a document or application) trigger the correct API calls and update the UI accordingly.
        - Resolve any UI inconsistencies or issues, ensuring a seamless user experience.

- ● **August 30 (Friday): Full Frontend-Backend Integration**
    - ○ **Task:** Integrate frontend with backend APIs for full functionality, including volunteer application.
        - Complete the integration of all frontend components with their corresponding backend APIs, ensuring full application functionality.
        - Perform end-to-end testing to validate that the entire application workflow (from login to document submission to volunteer application) works as expected.
        - Address any final issues or bugs identified during testing to ensure a smooth user experience.