**

**External Design Document (EDD)**

**Incident Management App**

***Team Alpha***

**Table of Contents**

1. [Project Overview](#project-overview)
2. [Wireframes](#wireframes)
   * [Login and Registration Page](#login-and-registration-page)
   * [Dashboard](#dashboard)
   * [Incident Management (CRUD)](#incident-management-crud)
3. [Initial Screenshots](#initial-screenshots)
4. [External Design Details](#external-design-details)
   * [Frontend](#frontend)
   * [Backend](#backend)
   * [API Design](#api-design)

**1. Project Overview**

**Application Name:** Incident Management App

**Purpose:**  
An application designed to help users efficiently manage and track incidents. It allows users to:

* Create, update, and delete incidents.
* Authenticate and securely log in to the app.
* View a user-friendly dashboard summarizing incident information.

**Technology Stack:**

* **Frontend:** React (Vite setup)
* **Backend:** Node.js, Express.js
* **Database:** MongoDB (Atlas)
* **Authentication:** JSON Web Tokens (JWT)

**2. Wireframes**

**2.1 Login and Registration Page**

* A simple form with fields for email, password, a “Login” and "Register" link.
* A validation message for incorrect credentials or registration errors.  
  *(Wireframe Example Placeholder)*

**2.2 Dashboard**

* Displays a summary of incidents (e.g., counts of open, in-progress, or resolved incidents).
* Navigation bar with links to Dashboard, Manage Incidents, and Profile.  
  *(Wireframe Example Placeholder)*

**2.3 Incident Management (CRUD)**

* **Create Incident Page:** A form to add incident details (title, description, priority, status).
* **Update Incident Page:** Similar to create, pre-filled with existing data.
* **Incident List:** A table showing all incidents with options to update or delete.  
  *(Wireframe Example Placeholder)*

**3. Initial Screenshots**

1. **Login Page** – Screenshot placeholder.
2. **Dashboard** – Screenshot placeholder.
3. **Incident Management Page** – Screenshot placeholder.

**4. External Design Details**

**Frontend**

* **Framework/Tools:** React.js, Vite, CSS
* **Features:**
  + Responsive UI for seamless use across devices.
  + Form validation for inputs (e.g., login credentials, incident forms).
  + Intuitive navigation for improved user experience.

**Backend**

* **Framework/Tools:** Node.js, Express.js
* **Features:**
  + RESTful APIs to handle all user and incident operations.
  + Middleware for request validation and security (e.g., rate-limiting, JWT authentication).

**API Design**

| **HTTP Method** | **Endpoint** | **Description** | **Authentication Required** |
| --- | --- | --- | --- |
| POST | /api/users/register | Register a new user | No |
| POST | /api/users/login | Authenticate user and return JWT | No |
| GET | /api/incidents | Fetch all incidents for the user | Yes |
| POST | /api/incidents | Create a new incident | Yes |
| PUT | /api/incidents/ | Update an existing incident | Yes |
| DELETE | /api/incidents/ | Delete an incident | Yes |