



External Design Document (EDD)

Incident Management App

Team Alpha

Sadik Elahi - 301303846

Table of Contents

Table of Contents

1. Project Overview
 2. Wireframes
 - Login and Registration Page
 - Dashboard
 - Incident Management (CRUD)
 - Interface Functionality
 3. Initial Screenshots
 4. External Design Details
 - Frontend
 - Backend
 - API Design
 - Docker and Deployment
 5. Environment Setup and Configuration
-

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.js, Vite
 - **Backend:** Node.js, Express.js
 - **Database:** MongoDB (Atlas)
 - **Authentication:** JSON Web Tokens (JWT)
 - **Deployment:** Docker, Google Cloud Run
-

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)

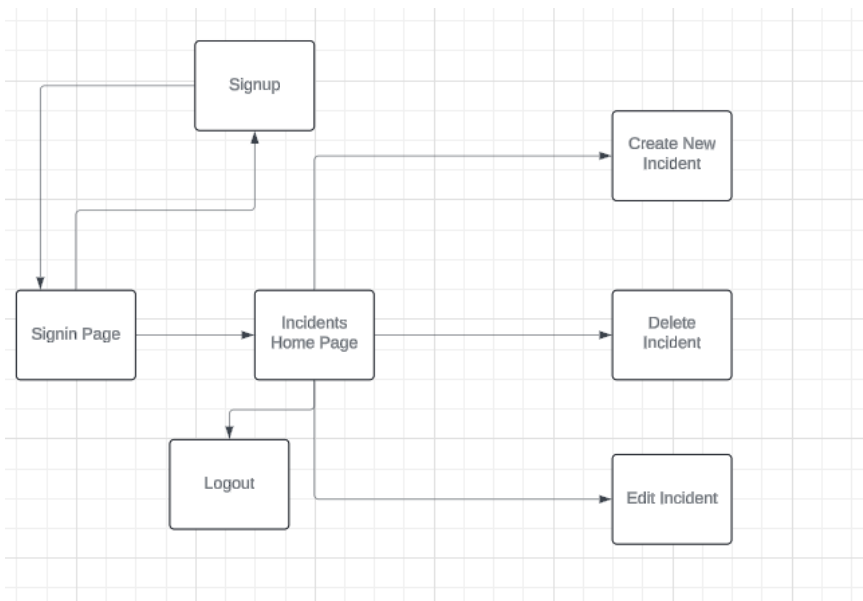
2.2 Dashboard

- 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)

2. Interface Functionality



3. Initial Screenshots

1. Login Page –



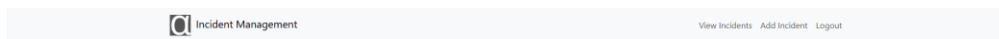
Sign In

Email

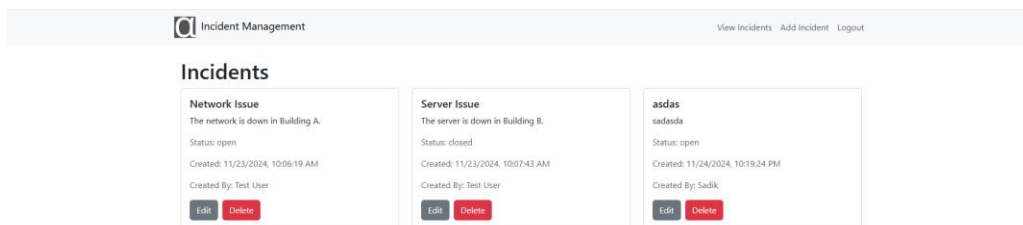
Password

Sign In

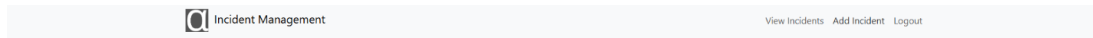
2. Navbar –



3. Incidents Page –



4. Create Incidents Page –



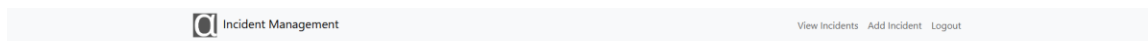
Create Incident

Title

Description

[Create Incident](#)

5. Edit Incident Page –



Edit Incident

Title

Description

Status

[Save Changes](#)

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).

Docker and Deployment

The backend and frontend of the Incident Management app are containerized using Docker, which allows for easy deployment across different environments.

- **Frontend:** The React app is containerized in a Docker image, tagged as frontend-image. It is deployed to Google Cloud Run, making it easily scalable and accessible.
- **Backend:** The Node.js backend is also containerized using Docker and deployed to Google Cloud Run with a backend service URL:
<https://backend-service-1091173744770.us-east1.run.app>.
- **Deployment Flow:**
 - The frontend communicates with the backend via API requests.
 - Both the frontend and backend containers are hosted on Google Cloud Run, ensuring scalability and resilience.

Docker Commands:

- **Build the Frontend Docker Image:**

docker buildx build -t frontend-image .

- **Build the Backend Docker Image:**

docker buildx build -t backend-image .

- **Push the Frontend Docker Image to Google Container Registry:**

docker push gcr.io/august-strata-444321-t2/frontend-image

- **Push the Backend Docker Image to Google Container Registry:**

```
docker push gcr.io/august-strata-444321-t2/backend-image
```

- **Deploy the Frontend to Google Cloud Run:**

```
gcloud run deploy frontend-service \
```

```
--image gcr.io/august-strata-444321-t2/frontend-image \
```

```
--platform managed \
```

```
--region us-east1 \
```

```
--allow-unauthenticated
```

- **Deploy the Backend to Google Cloud Run:**

```
gcloud run deploy backend-service \
```

```
--image gcr.io/august-strata-444321-t2/backend-image \
```

```
--platform managed \
```

```
--region us-east1 \
```

```
--allow-unauthenticated
```

Backend URL for Frontend:

The frontend app communicates with the backend service via the backend URL:

<https://backend-service-1091173744770.us-east1.run.app>

5. Environment Setup and Configuration

Backend Environment Setup:

- **Database Connection:** The backend uses MongoDB Atlas as its database provider. The connection URL is provided via an .env file:

```
MONGO_URI="mongodb+srv://your_mongo_connection_string"
```

```
JWT_SECRET="your_jwt_secret_key"
```

```
PORT=3000
```


Frontend Configuration:

- **API Base URL:** The React app communicates with the backend via Axios. The baseURL for Axios should be updated to point to the backend URL deployed on Google Cloud Run:

```
const api = axios.create({  
  
  baseURL: 'https://backend-service-1091173744770.us-east1.run.app', // Updated  
  Backend URL  
  
});
```

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