

# Incident Management System Documentation

## Submitted By

Mohammad Sazid – 930298

Pranav Marwaha – 930271

Vaibhav Bhargava – 930281

## Project Overview

The Incident Management System is a web application designed to efficiently manage user queries.

The system enables users to report incidents, view their status, and track their progress. This application leverages Angular for the frontend, Spring Boot for the backend, and MySQL for data management.

## Technologies Used

**Frontend:** Angular

**Backend:** Springboot

**Database:** MySQL

## Installation and Setup

### Application Ports:

Frontend: Running on port 4200.

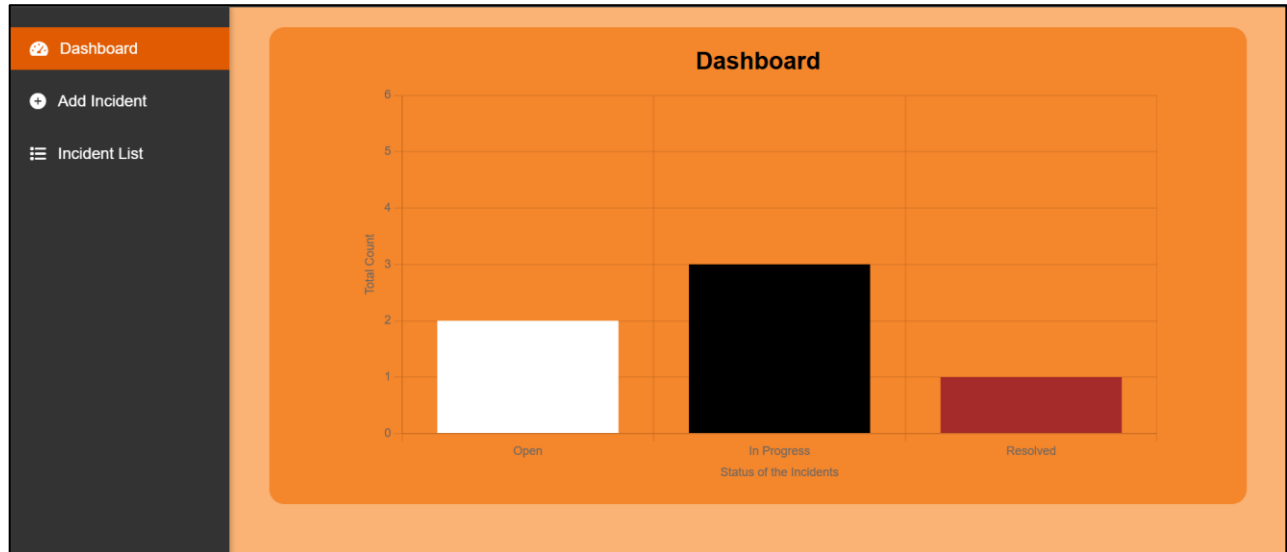
Backend: Running on port 8080.

Database: The incidents are stored under the table incidents in the hibernate\_db database.

## Frontend

### 1. Dashboard

Displays a bar graph summarizing the status of incidents (Open, In Progress, Resolved)



### 2. Add Incident

- Users can submit incidents through a form that captures essential details, including the incident title, description, priority, and status.
- If the details are completely filled before hitting the create button, a notification is dropped down stating all fields be filled before the said action can take place.
- Additionally, the date and time of creation of an incident will be automatically recorded on the backend.

Classification: Internal  
Classification: Internal

Dashboard

Add Incident

Incident List

Create Incident

Title:

Enter Title

Status:

Please choose

Description:

Enter Description

Priority:

Low

Create

Dashboard

Add Incident

Incident List

Please enter all fields.

Title:

example incident

Status:

Open

Description:

Enter Description

Priority:

Low

Create

Dashboard

Add Incident

Incident List

Incident has been created!

Title:

Enter Title

Status:

Please choose

Description:

Enter Description

Priority:

Create

### 3. Incident List

- Lists all incidents created and stored in the database.
- Provides the functionality to edit and delete an incident.
- Incidents are sorted based on the order of their priority.

[Dashboard](#)  
[Add Incident](#)  
[Incident List](#)

#### Incident List

id	Title	Description	Status	Priority	Created On	Actions
1	black screen	needs to be resolved urgently	In Progress	High	30/10/24 03:28 PM	<a href="#">Edit</a> <a href="#">Delete</a>
2	keyboard problem	keys not working	In Progress	High	30/10/24 03:28 PM	<a href="#">Edit</a> <a href="#">Delete</a>
7	machine failure	machine failed to boot up	Resolved	High	30/10/24 05:50 PM	<a href="#">Edit</a> <a href="#">Delete</a>
3	mysql installation	need software on system	Open	Medium	30/10/24 03:28 PM	<a href="#">Edit</a> <a href="#">Delete</a>
5	mssql issue	database not found	In Progress	Low	30/10/24 05:12 PM	<a href="#">Edit</a> <a href="#">Delete</a>
8	example incident	creating incident for documentation	Open	Low	30/10/24 06:05 PM	<a href="#">Edit</a> <a href="#">Delete</a>

[Dashboard](#)  
[Add Incident](#)  
[Incident List](#)

#### Incident List

id	Title	Description	Status	Priority	Created On	Actions
1	black screen	needs to be resolved urgently	In Progress	High	30/10/24 03:28 PM	<a href="#">Edit</a> <a href="#">Delete</a>
2	keyboard problem	keys not working	In Progress	High	30/10/24 03:28 PM	<a href="#">Edit</a> <a href="#">Delete</a>
7	machine failure	machine failed to boot up	Resolved	High	30/10/24 05:50 PM	<a href="#">Edit</a> <a href="#">Delete</a>
3	mysql installation	need software on system	Open	Medium	30/10/24 03:28 PM	<a href="#">Edit</a> <a href="#">Delete</a>
5	mssql issue	database not found	In Progress	Low	30/10/24 05:12 PM	<a href="#">Edit</a> <a href="#">Delete</a>
8	example incident	creating incident for documentation	Open	Low	30/10/24 06:05 PM	<a href="#">Edit</a> <a href="#">Delete</a>

Confirm Delete

Are you sure you want to delete this incident?

[Cancel](#) [Delete](#)

Classification: Internal  
Classification: Internal

Dashboard

Add Incident

Incident List

Id	Title	Priority	Created On	Actions
1	black screen	High	30/10/24 03:28 PM	Edit Delete
2	keyboard pro	High	30/10/24 03:28 PM	Edit Delete
7	machine failu	High	30/10/24 05:50 PM	Edit Delete
3	mysql installa	Medium	30/10/24 03:28 PM	Edit Delete
5	mssql issue	Low	30/10/24 05:12 PM	Edit Delete
8	example inci	Low	30/10/24 06:05 PM	Edit Delete

Edit Incident

Title: mssql issue

Status: In Progress

Description: database not found

Priority: Low

Cancel

Submit

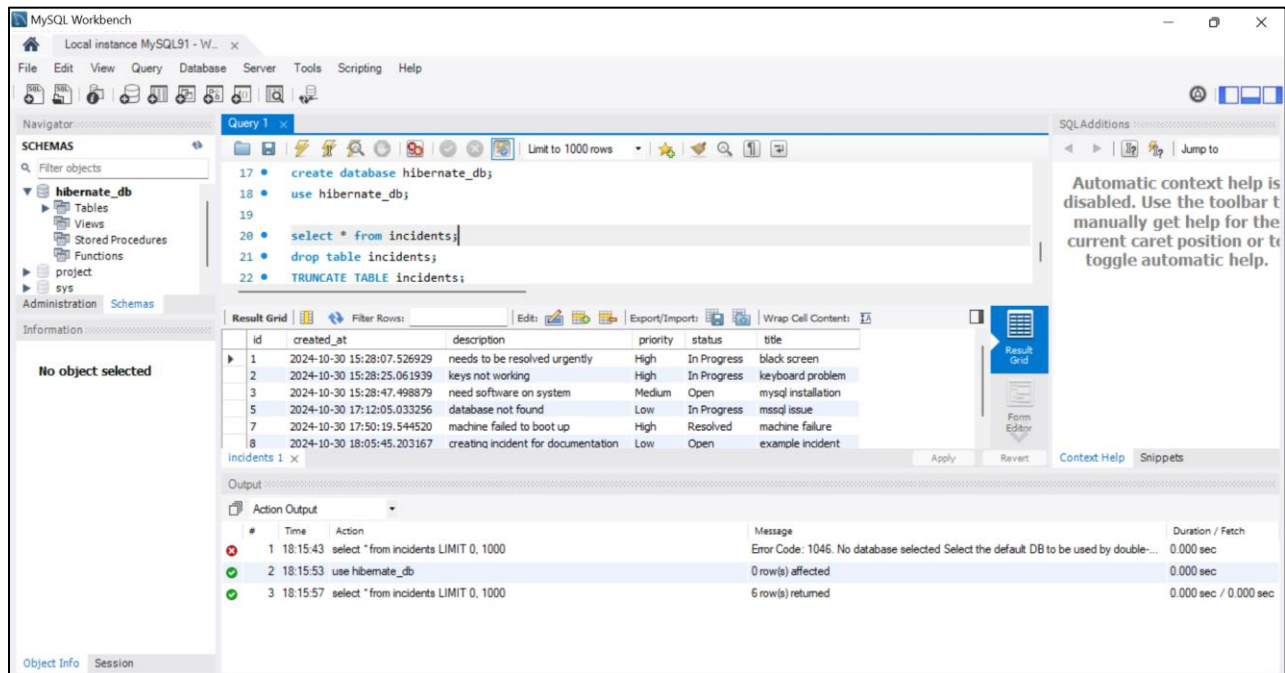
## Backend

```
Project Explorer
└─ Final_Project_Backend (in backend)
   └─ src/main/java
      └─ com.example.Final_Project_Backend
         └─ IncidentController.java

IncidentController.java
21 return incidentService.getAllIncidents();
22 }
23
24 @PostMapping("/add-incident")
25 public ResponseEntity<Incident> createIncident(@RequestBody Incident incident) {
26     Incident createdIncident = incidentService.createIncident(incident);
27     return ResponseEntity.ok(createdIncident);
28 }
29
30 @DeleteMapping("/delete/{id}")
31 public void deleteIncident(@PathVariable Long id) {
32     incidentService.deleteIncident(id);
33 }
34
35 @PutMapping("/update/{id}")
36 public void updateIncident(@PathVariable Long id, @RequestBody Incident incident) {
37     incidentService.updateIncident(id, incident);
38 }

Console
FinalProjectBackendApplication [Java Application] C:\Users\ms930298\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.22.0.2\v20240802-1626\jre\bin\java.exe
5 --- [Final_Project_Backend] [ restartedMain] jpaBaseConfigurationsJpaWebConfiguration : spring.jpa.open-in-view is enabled by de
6 --- [Final_Project_Backend] [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 357
6 --- [Final_Project_Backend] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with
6 --- [Final_Project_Backend] [ restartedMain] c.e.F.FinalProjectBackendApplication : Started FinalProjectBackendApplication i
6 --- [Final_Project_Backend] [ restartedMain] .ConditionEvaluationDeltaLoggingListener : Condition evaluation unchanged
6 --- [Final_Project_Backend] [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'c
6 --- [Final_Project_Backend] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
6 --- [Final_Project_Backend] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms
t,ii_0.description,ii_0.priority,ii_0.status,ii_0.title from incidents ii_0
t,ii_0.description,ii_0.priority,ii_0.status,ii_0.title from incidents ii_0
t,ii_0.description,ii_0.priority,ii_0.status,ii_0.title from incidents ii_0 where ii_0.id=?

Problems 0 errors, 5 warnings, 0 others
```



## Responsibilities

- **Frontend Development – done by Pranav Marwaha**
  - Developed three components:
    - **Add Incident:** Developed a user-friendly component that allows users to submit new incidents easily, complete with built-in validators to ensure data integrity.
    - **Edit Incident:** Created a flexible interface that allows users to seamlessly modify existing incident details, ensuring that changes are effectively reflected in the database.
    - **Confirm Dialog Box:** Implemented a dialog box to prompt users for confirmation before deleting an incident.
- **Frontend Development and Database Connection – done by Vaibhav Bhargava**
  - Developed two components:
    - **Incident List:** Displays incidents with various fields, including the automatically generated date and time of creation from the backend. It also provides options to edit or delete incidents as needed.
    - **Dashboard:** Visualizes data through a bar graph using the Chart.js library, offering an insightful overview of incidents.

- Database Connection:
  - Connected to MySQL database.
- **Backend Development and Frontend Styling – done by Mohammad Sazid**
  - Backend Development:
    - The code is well-structured, with files organized in a modular and systematic manner.
    - Code is divided between several packages such as controllers, entities, services, and repositories.
  - Frontend Styling:
    - Designed and implemented the styling for all frontend components, ensuring a cohesive visual experience that stays synchronized with the data received.

## Screens

### 1. Dashboard

- **Functionality:** Offers a visual representation of the statuses of all incidents.
- **Bar Graph:** A bar graph showing:
  - Open: Number of incidents in the Open status.
  - In Progress: Number of incidents currently being worked on.
  - Resolved: Number of incidents that have been resolved.

### 2. Create Incident

- **Functionality:** Users can add new incidents.
- **Form fields:**
  - **Title:** Subject of the incident.
  - **Description:** Brief explanation of the issue.
  - **Status:** Dropdown to select the current status of incident (Open, In Progress, Resolved).
  - **Priority:** Dropdown to select the priority level of incident (Low, Medium, High).

### 3. Incident List

- **Functionality:** Displays all incidents in a tabular format.
- **Features:**
  - User can view all incidents stored in database.
  - User can edit any incident at will effectively making change in database.
  - User can delete any incident at will effectively making change in database.

### Conclusion

The Query Management System effectively enables users to manage incidents with a user-friendly interface, leveraging the power of Angular for the frontend, Spring Boot for the backend, and MySQL for data storage.