

Patrol tracking App and Web-application



Mentor:
Sravan Kumar(General manager)

By:
P.Rohan Reddy

Introduction

This document outlines both, functional and Non-functional requirements of Patroller Tracking app and web application

This Document basically gives a detailed overview of the working of the software product and how it is expected to perform. The document describes the product's user interface, hardware and software requirements.

Project Description

This is a pilot project for automating Work force Management of the Patrollers of a telecom broadband service organization. The patrollers perform a daily verification of the Fiber running across the locations to proactively monitor the well-being of the Fiber and also the hygiene of the environments (Fiber ducts, etc).

The pilot project considers an organization of 500 employees with 20 managers covering Fiber lines of about 50 kms.

Each of these fiber lines or spans(distance allocated) must be thoroughly checked every day so that customers get a seamless experience of the internet. Everyday each patroller goes and checks if the fiber line is secure or not and updates a checklist given to him . At the end of day the employee submits the report of task lists to the manager. The Objective of this project is to automate the above activity through a gps based mobile app and administration by Managers through a web app.

Solution

Mobile App : Used by Patrollers to login, accept their daily schedule, go through the route allocated to them and fill the checklist. Web App : Used by Managers to assign the routes to the Patrollers, track the Patrollers through the website and manage the issues raised by Patrollers through the checklist.

User Categories and Privileges

1. Manager

The Manager would be using the website to check if the employee is doing his work or not

2. Employee

The employee will use the app to complete the task list and to track his location

Requirements

- **Data Management and Scheduling**

Manager should be able to manage data i.e employee management and should also be able to schedule the fiber line route to the employee

- **Optimized data storage system**

Storing Data in an optimized way is important as we are storing the locations of employee his basic account information and his route schedule.

- **Implementing a user-friendly and stable software**

The employee or manager desires ease of access. We thus need to create an software and a user interface that is easy to use and doesn't crash or have glitches

- **Access on multiple devices**

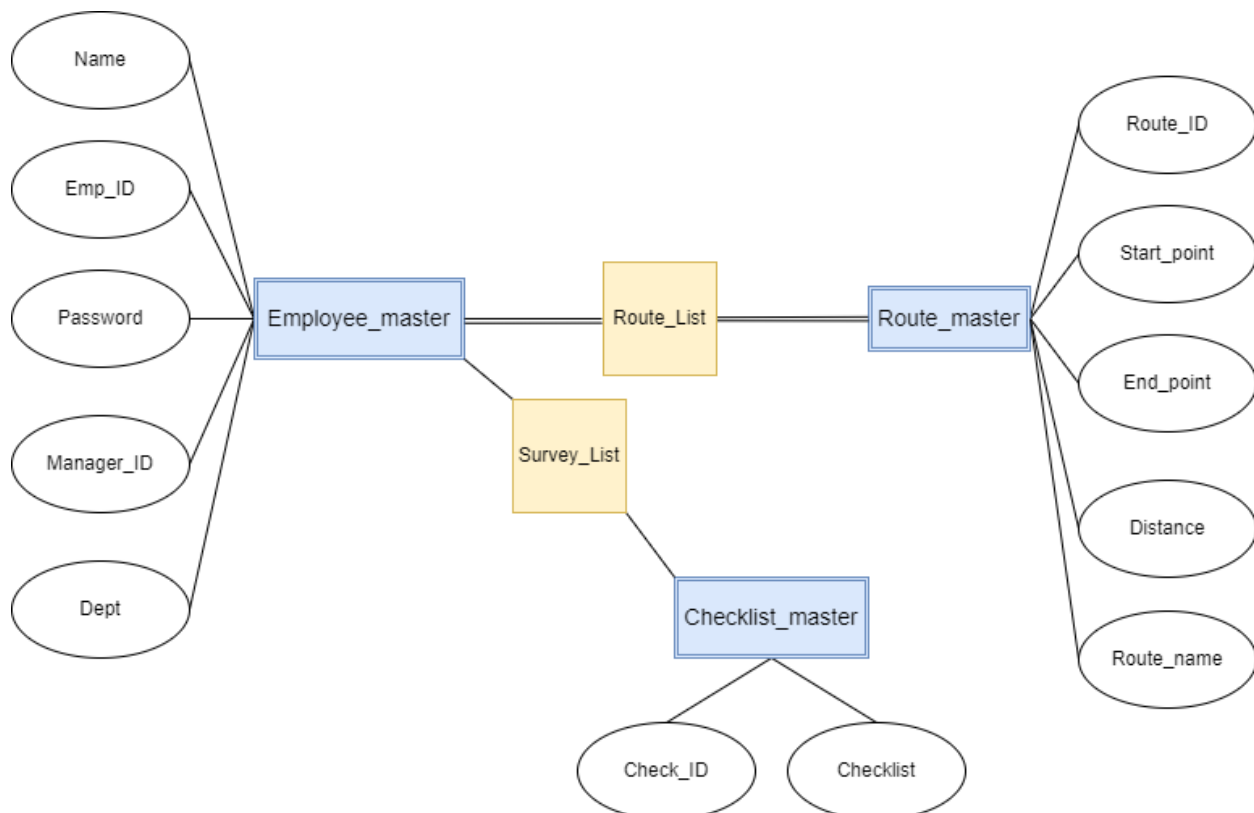
The Tracking application should be able to work on multiple devices

Other Requirements

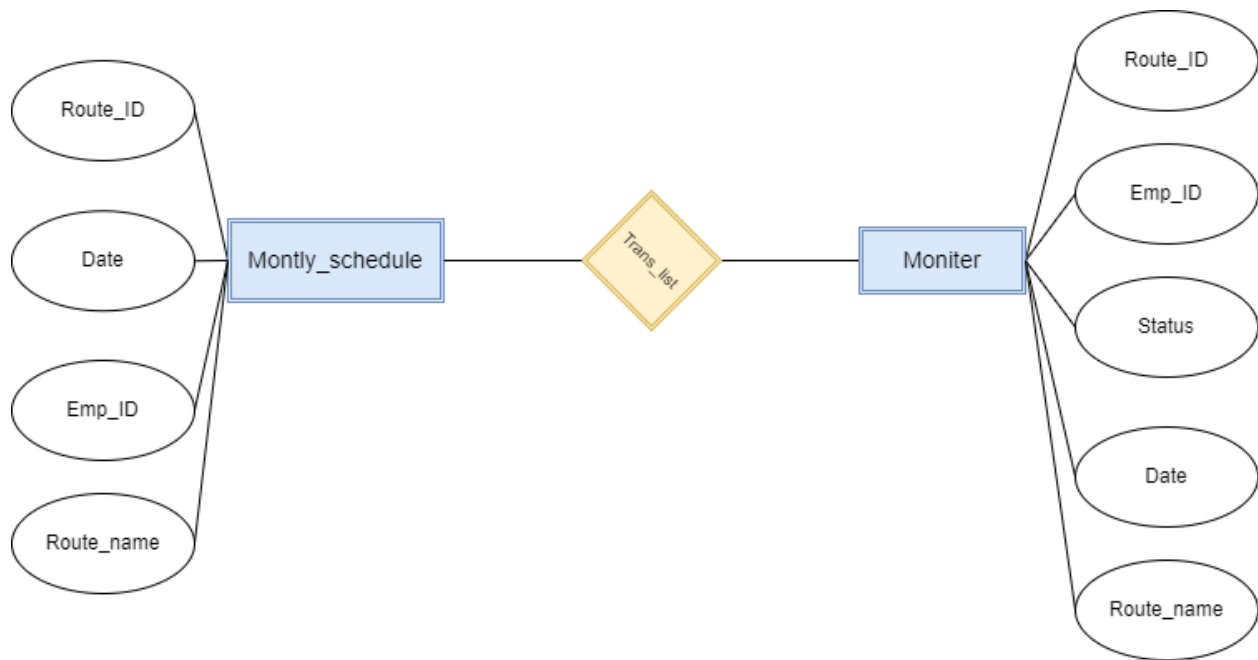
- Database should be designed in such a way that it can be scaled according to the number of users.
- System should ensure consistency and integrity of the user's transactions.
- It should be possible to monitor the database from time to time.
- Users should have complete control on their account and the ability to manage their account on their multiple devices

E-R diagrams

1).E-R Diagram-1



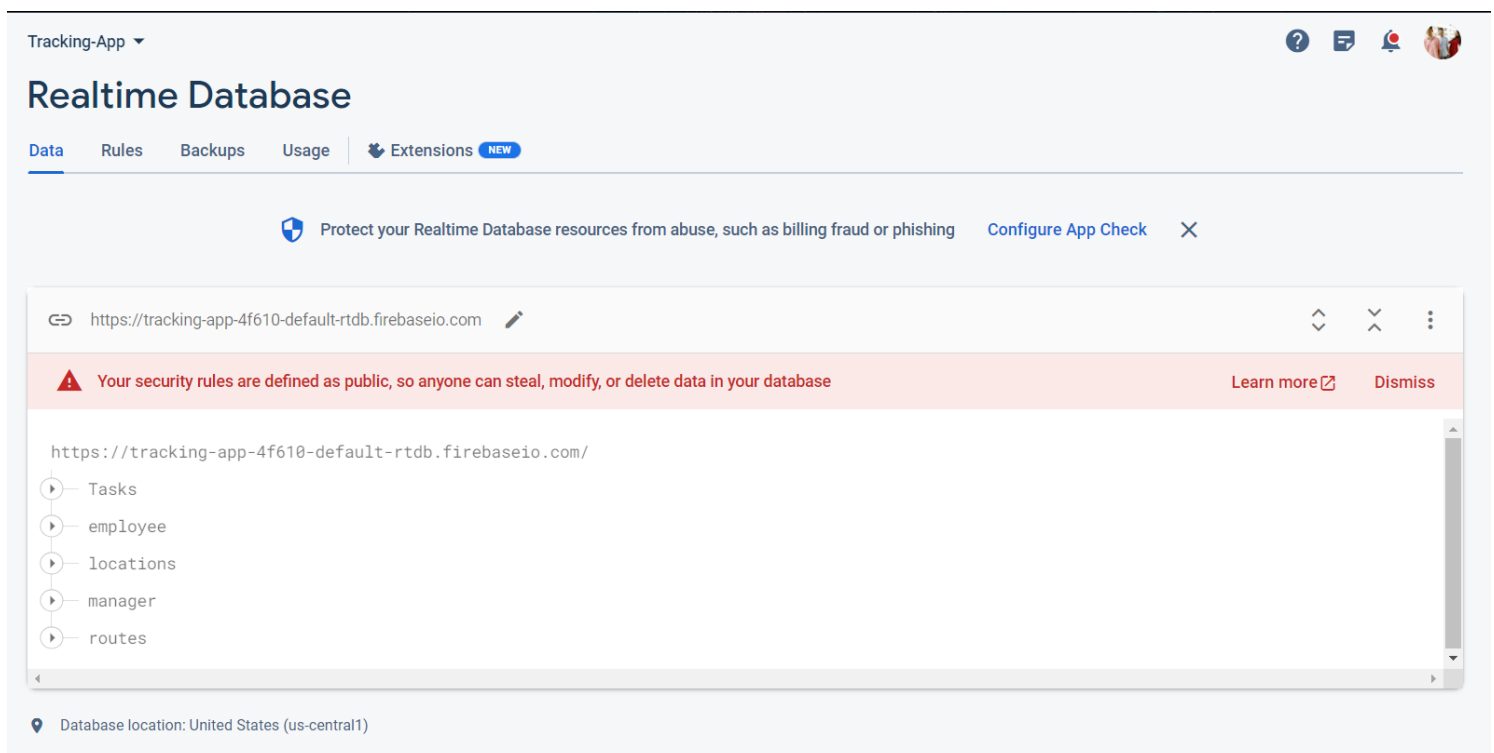
2).E-R Diagram-2(Transnational)



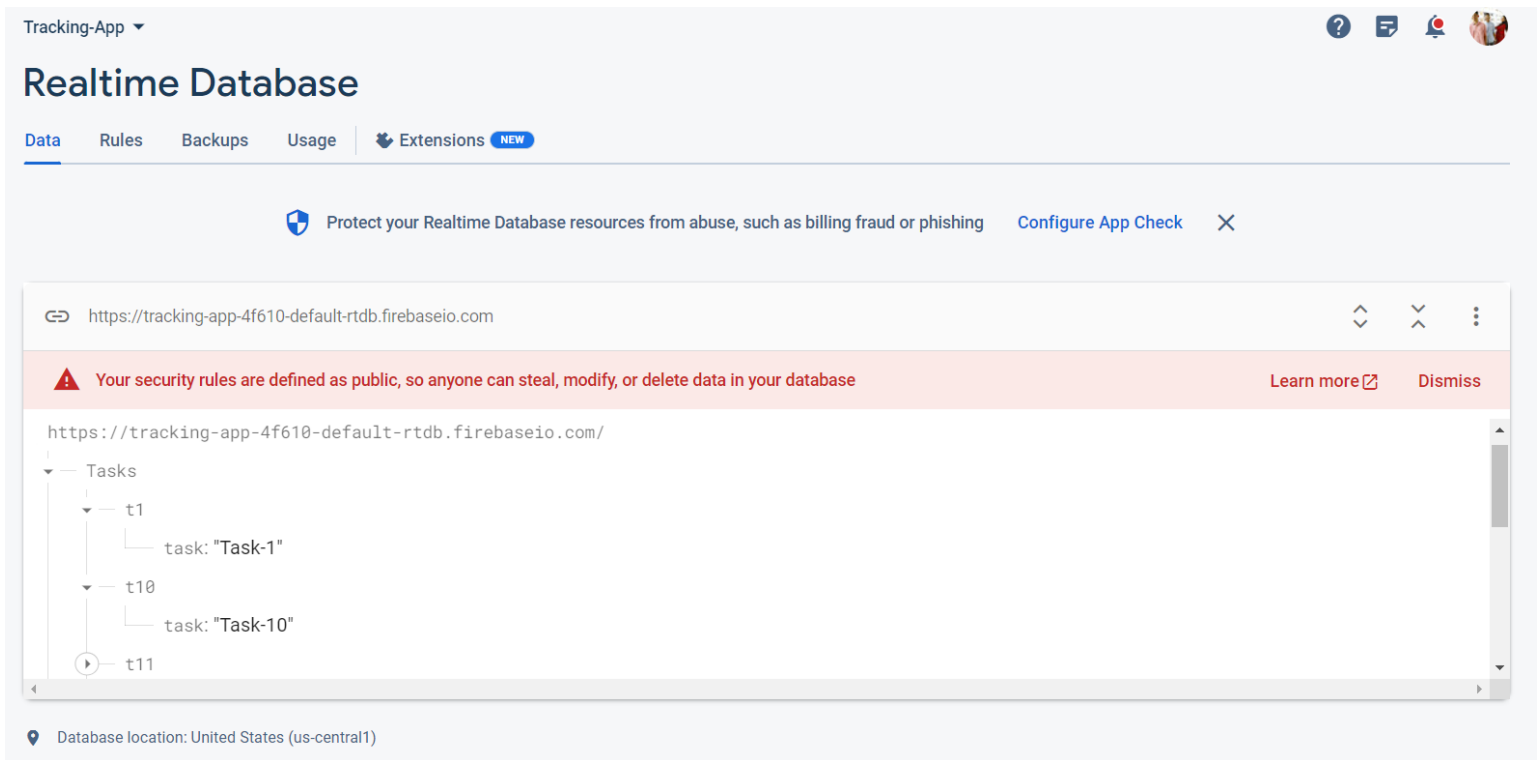
Database

In this development project I used firebase realtime database to store the data in the project.

Here are some snapshots of the database:-



1).Tasks contain the list to be checked by the employee on fiberline

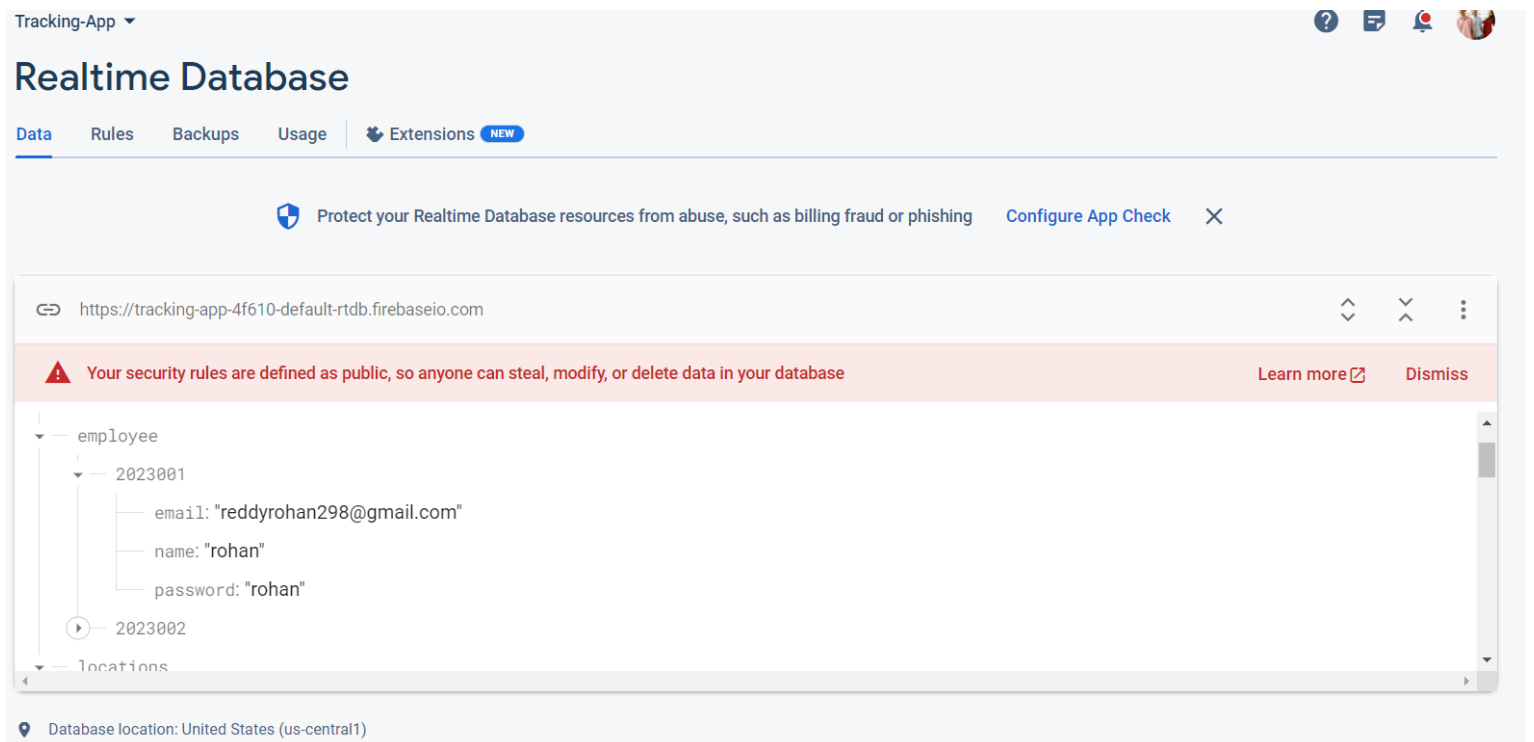


The screenshot shows the Firebase Realtime Database console for the 'Tracking-App'. The 'Data' tab is selected, displaying the database structure. A warning banner at the top states: 'Your security rules are defined as public, so anyone can steal, modify, or delete data in your database'. The database location is 'United States (us-central1)'. The 'Tasks' node is expanded, showing a list of tasks: 't1' (task: 'Task-1'), 't10' (task: 'Task-10'), and 't11'.

```
https://tracking-app-4f610-default-rtdb.firebaseio.com/

├── Tasks
│   ├── t1
│   │   └── task: "Task-1"
│   ├── t10
│   │   └── task: "Task-10"
│   └── t11
```

2).Employee contains Details of employee



The screenshot shows the Firebase Realtime Database console for the 'Tracking-App'. The 'Data' tab is selected, displaying the database structure. A warning banner at the top states: 'Your security rules are defined as public, so anyone can steal, modify, or delete data in your database'. The database location is 'United States (us-central1)'. The 'employee' node is expanded, showing details for two employees: '2023001' (email: 'reddyrohan298@gmail.com', name: 'rohan', password: 'rohan') and '2023002'.

```
https://tracking-app-4f610-default-rtdb.firebaseio.com/

├── employee
│   ├── 2023001
│   │   ├── email: "reddyrohan298@gmail.com"
│   │   ├── name: "rohan"
│   │   └── password: "rohan"
│   └── 2023002
└── locations
```

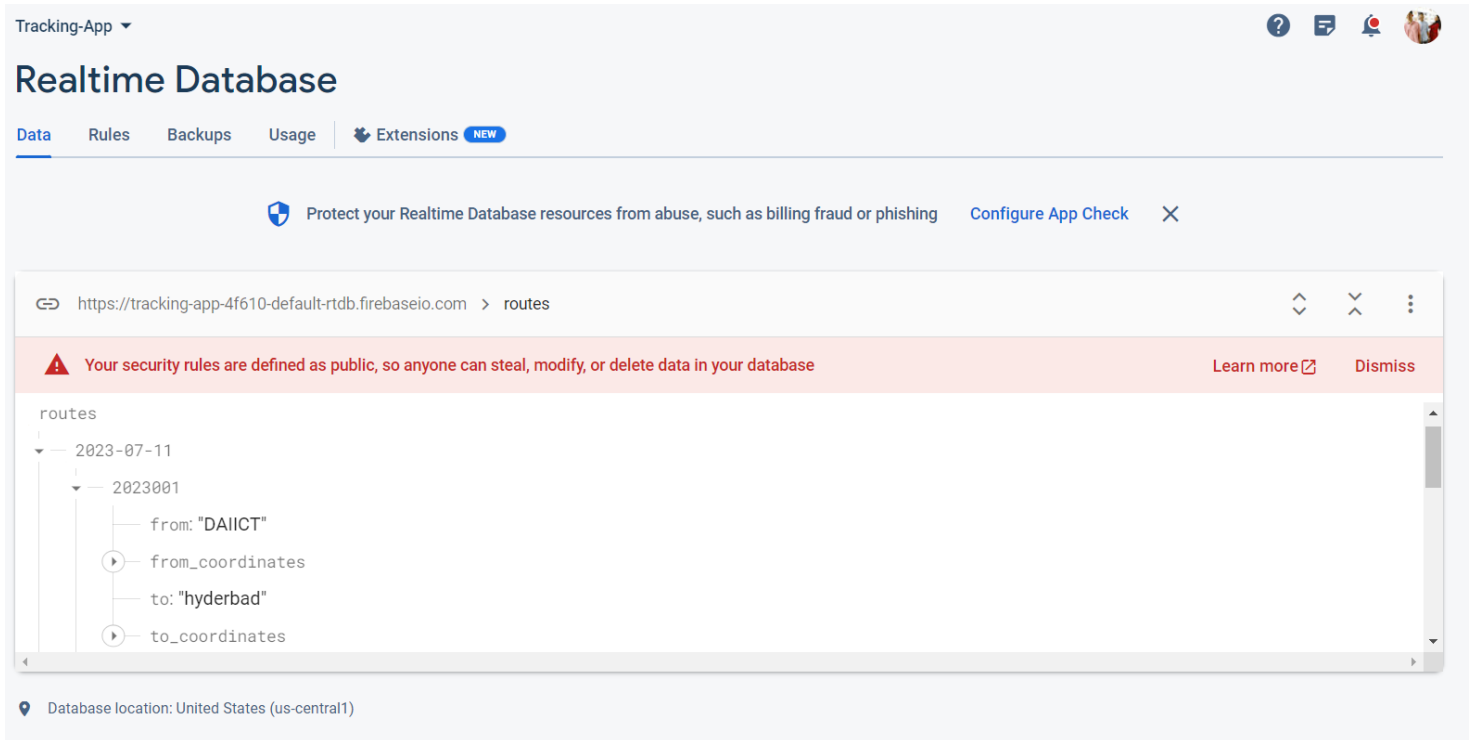
3).Locations contains latitude and longitude of the employee based on date,employee id and time

The screenshot shows the Firebase Realtime Database console for a project named 'Tracking-App'. The 'Data' tab is selected, displaying a tree view of the database structure. The path is: `locations` > `employee` > `2023001` > `2023-07-11` > `file2` > `-N_2Rqc105p55mEwHaQ5`. The value for the last node is `latitude: 23.1875217`. A warning banner at the top states: 'Your security rules are defined as public, so anyone can steal, modify, or delete data in your database'. The database location is 'United States (us-central1)'.

4).Manager contains details of the manager

The screenshot shows the Firebase Realtime Database console for the same 'Tracking-App' project. The 'Data' tab is selected, displaying a tree view of the database structure. The path is: `manager` > `202301`. The value for the `202301` node is a JSON object: `{email: "songoku2470@gmail.com", name: "rahul", password: "rahul"}`. A warning banner at the top states: 'Your security rules are defined as public, so anyone can steal, modify, or delete data in your database'. The database location is 'United States (us-central1)'.

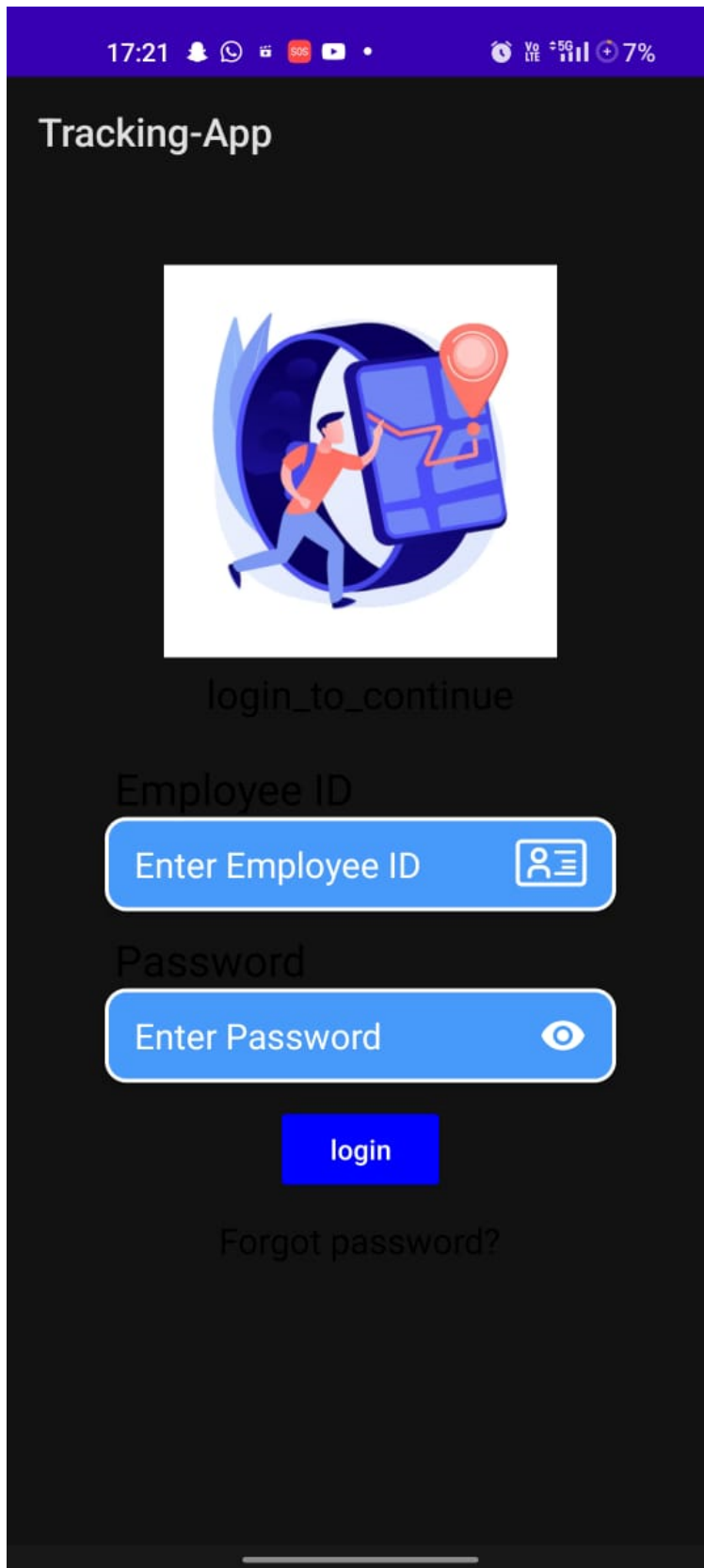
5).Routes contains schedule of the employee i.e based on date and employee id it from and to locations



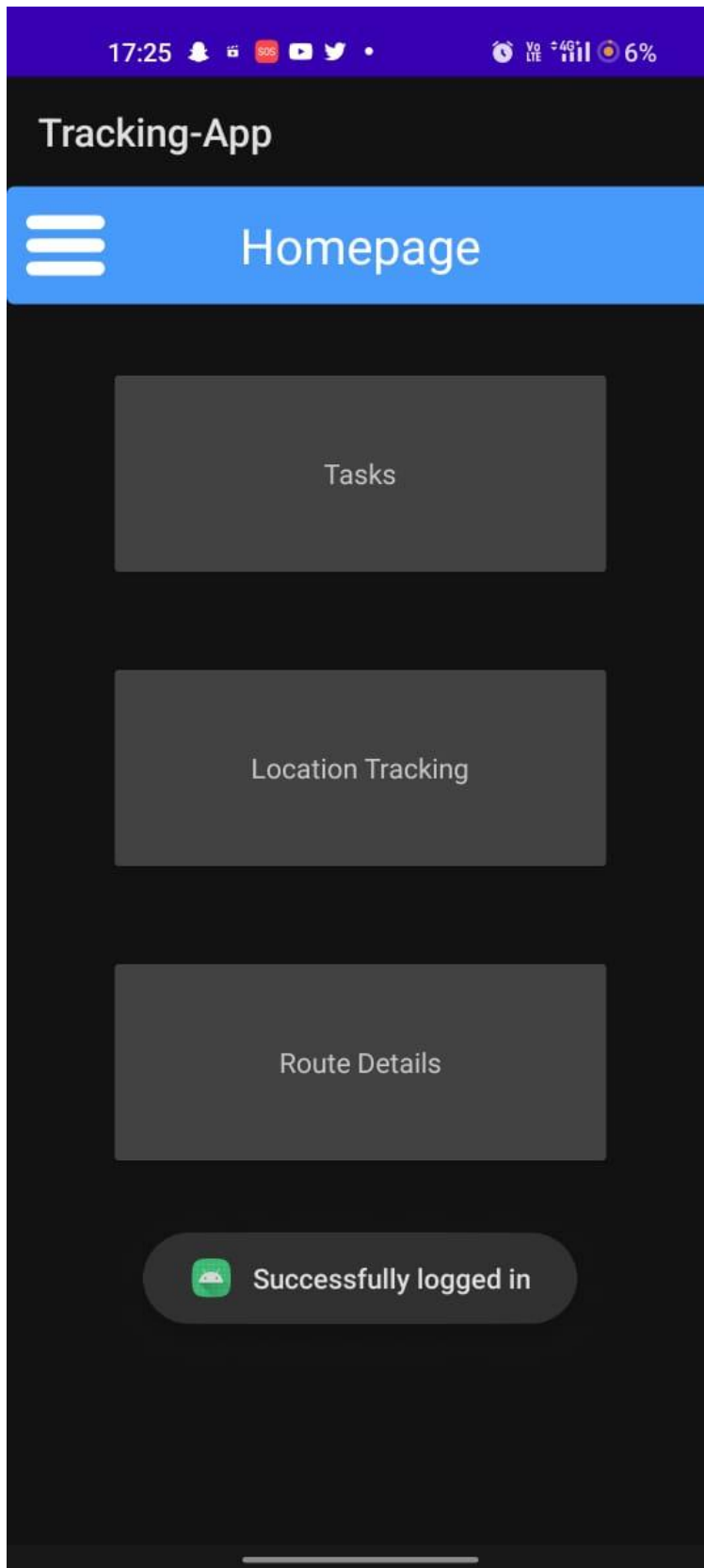
Tracking-App(Used by Employee)

I used android studio software for the development of the application used by the employee

Here are some snapshots of the app:-
1).Login Page:-



2)Home page:-



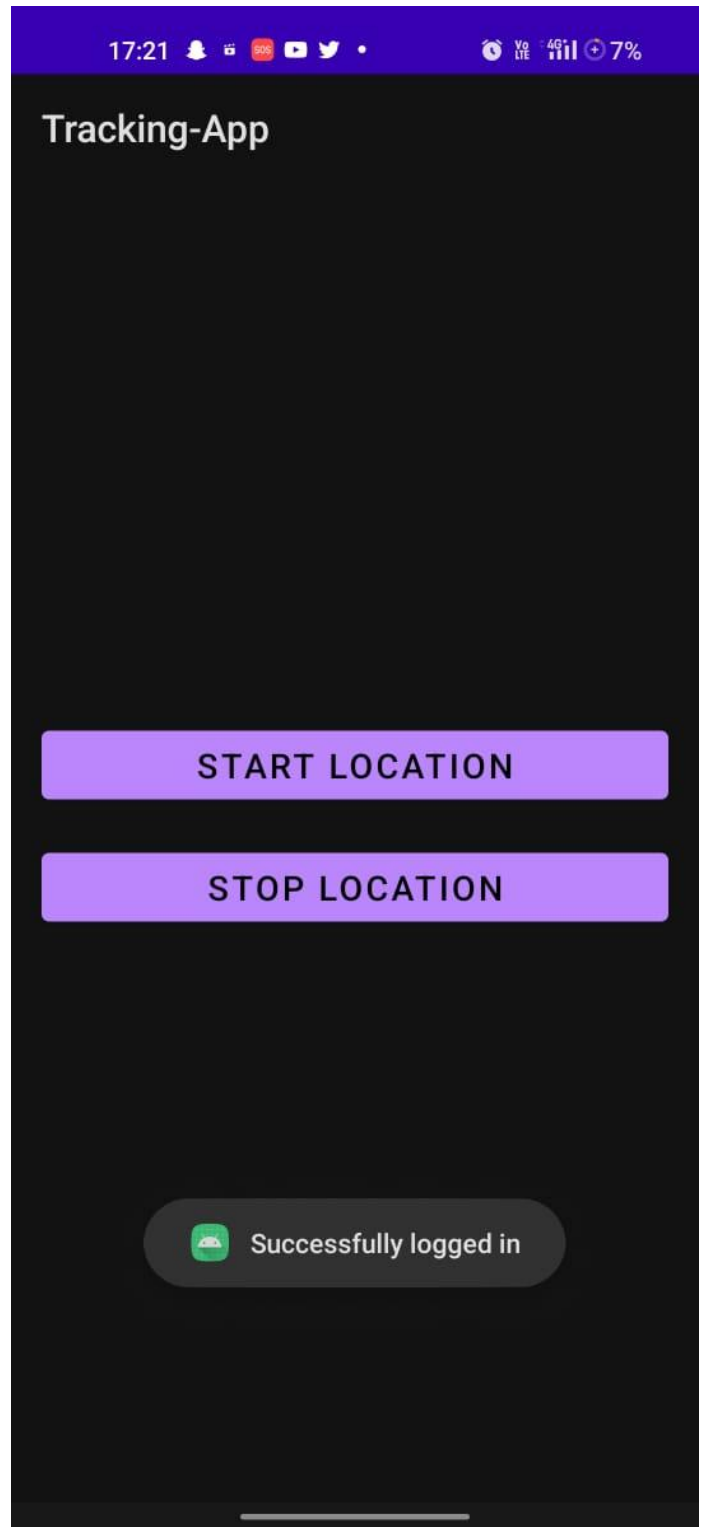
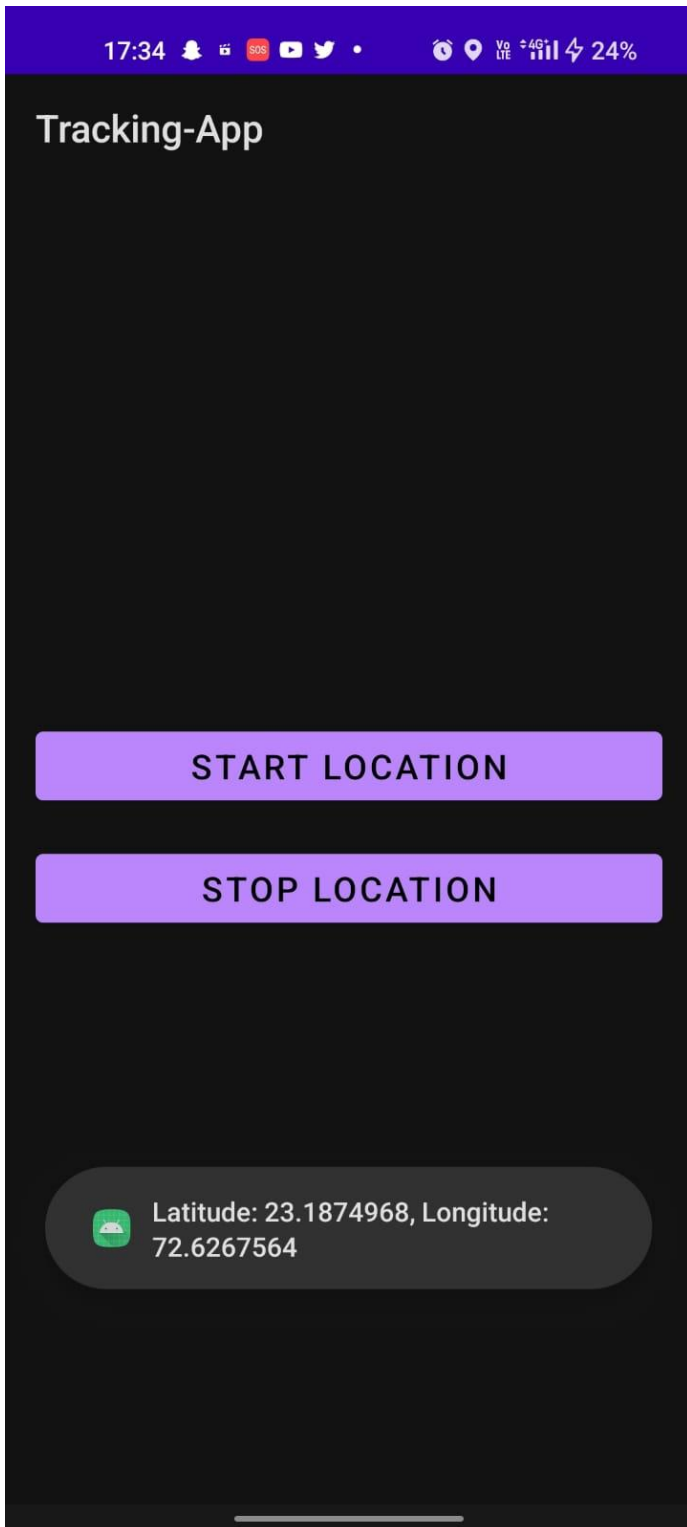
When we press Tasks button:-

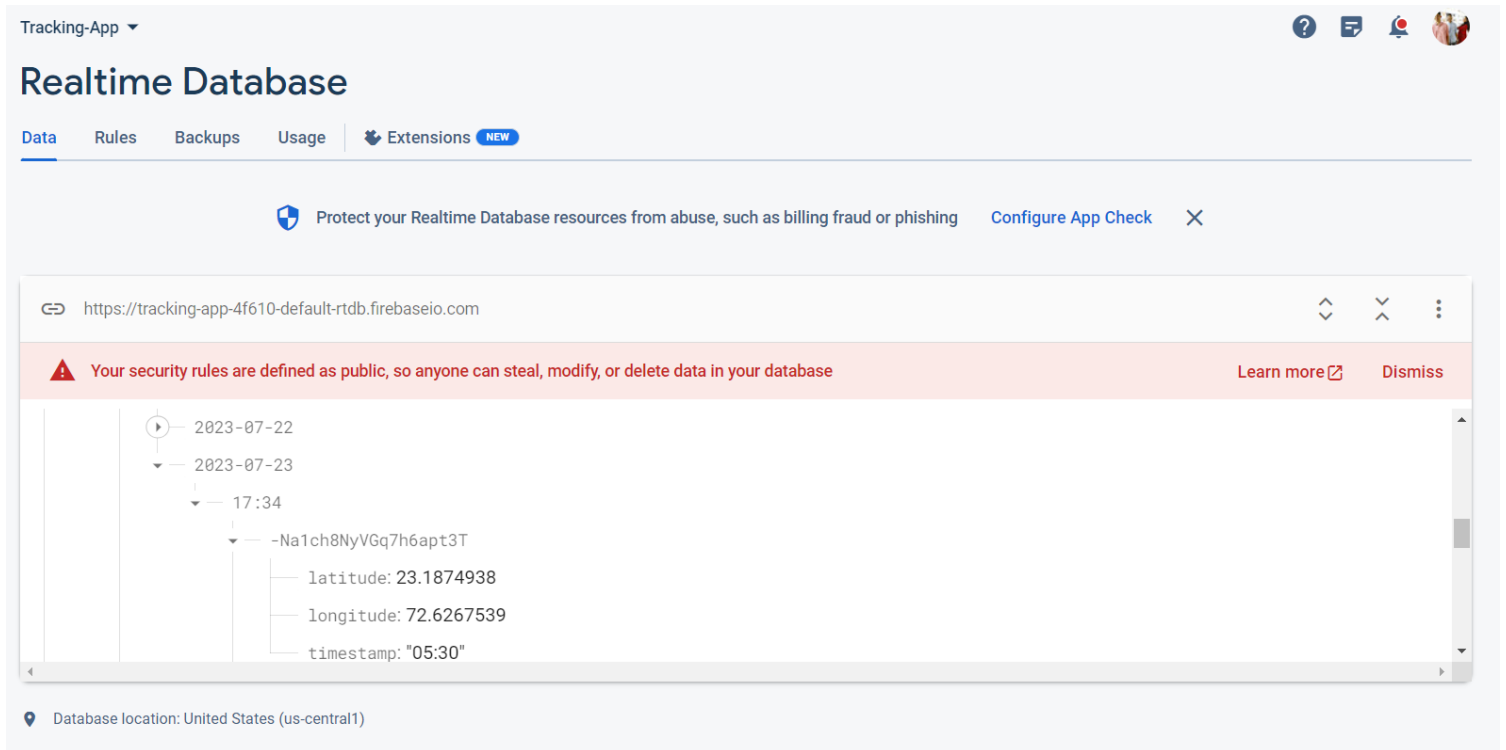
The screenshot shows a mobile application interface with a dark theme. At the top, a status bar displays the time 17:21, various social media icons, and battery status at 7%. Below the status bar, the app title 'Tracking-App' is visible. A blue header bar contains a white hamburger menu icon and the word 'Checklist'. The main content area lists eight tasks, each preceded by a black circle icon. Each task has two radio button options: 'Yes' (which is selected with a blue dot) and 'No' (which is unselected). At the bottom of the screen, there is a blue rounded rectangular button with the text 'Done'.

Task	Yes	No
Task-1	<input checked="" type="radio"/>	<input type="radio"/>
Task-2	<input checked="" type="radio"/>	<input type="radio"/>
Task-3	<input checked="" type="radio"/>	<input type="radio"/>
Task-4	<input checked="" type="radio"/>	<input type="radio"/>
Task-5	<input checked="" type="radio"/>	<input type="radio"/>
Task-6	<input checked="" type="radio"/>	<input type="radio"/>
Task-7	<input checked="" type="radio"/>	<input type="radio"/>
Task-8	<input checked="" type="radio"/>	<input type="radio"/>

Done

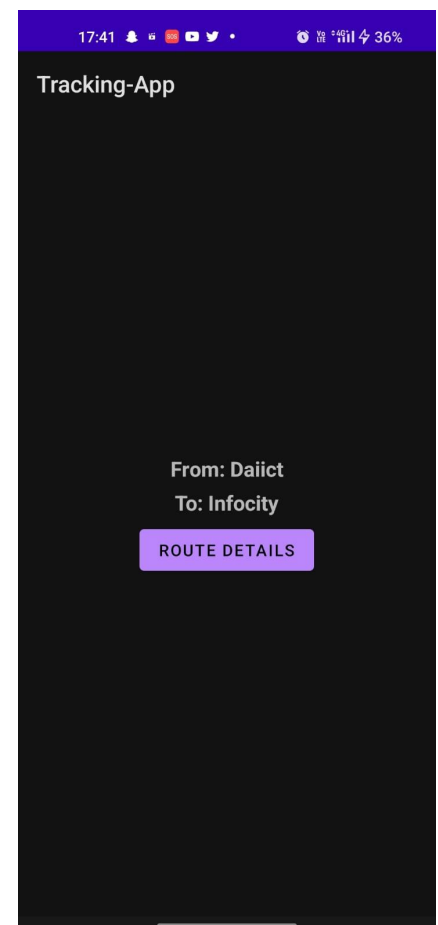
When we press location Tracking button and here when we press start location button it start to take the coordinates and store it in firebase realtime database :-





In above image we can see the the location is being stored based on date and time

When we press Route details Button we get:-

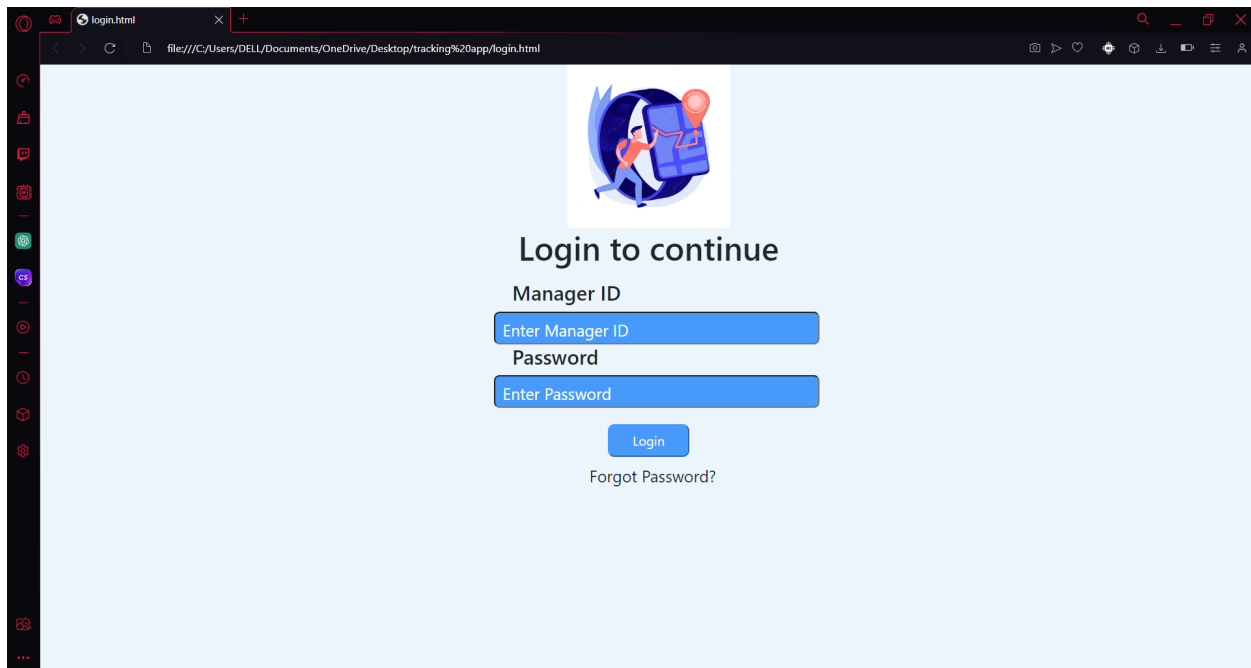


Tracking-Web application(Used by manager)

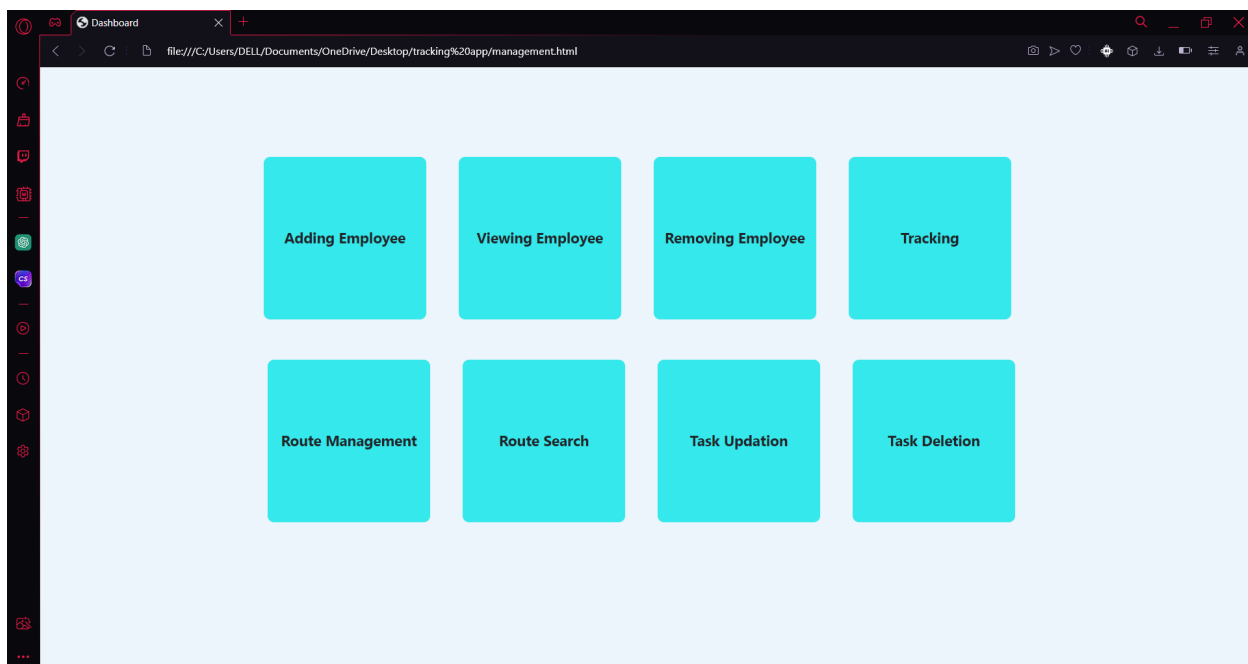
I used HTML,CSS,Javascript for the development of the website

Here are some snapshots of the website:-

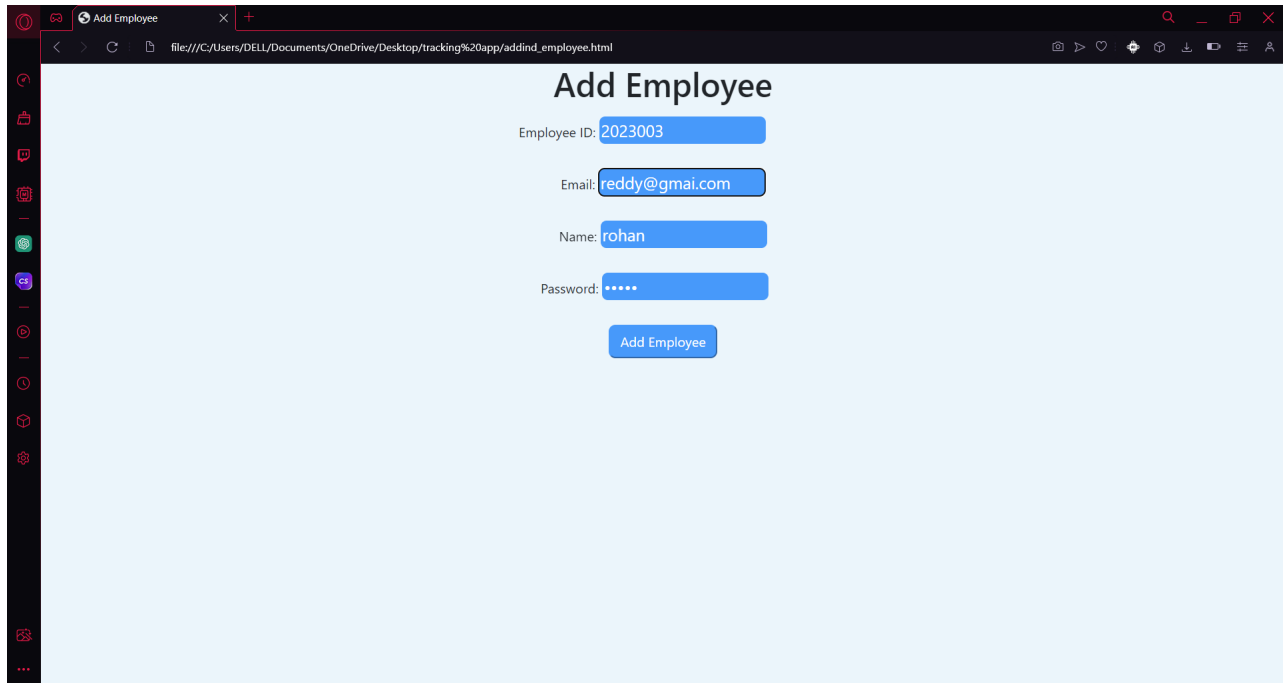
1).Login page:-



2).Home page:-



3).when we press Adding employee Button:-

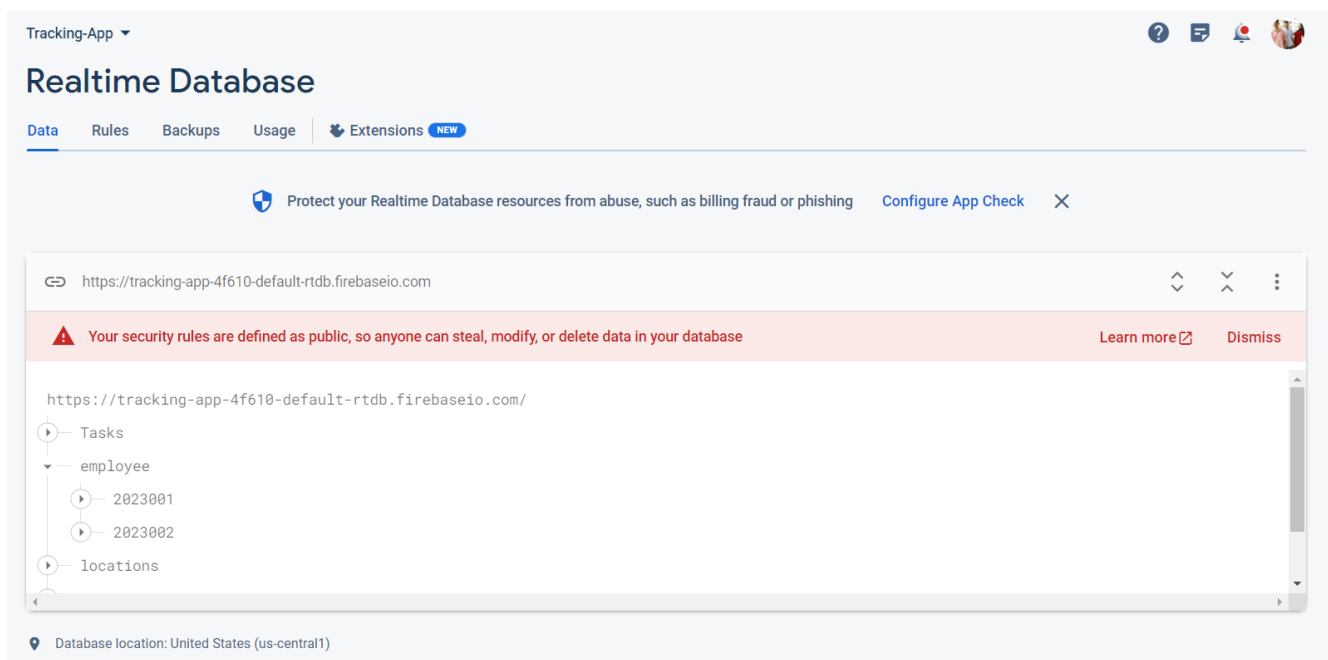


The screenshot shows a web browser window with the title 'Add Employee'. The address bar shows the file path: file:///C:/Users/DELL/Documents/OneDrive/Desktop/tracking%20app/addind_employee.html. The form is titled 'Add Employee' and contains the following fields:

- Employee ID: 2023003
- Email: reddy@gmail.com
- Name: rohan
- Password: *****

Below the fields is a blue button labeled 'Add Employee'.

In database before adding employee:-

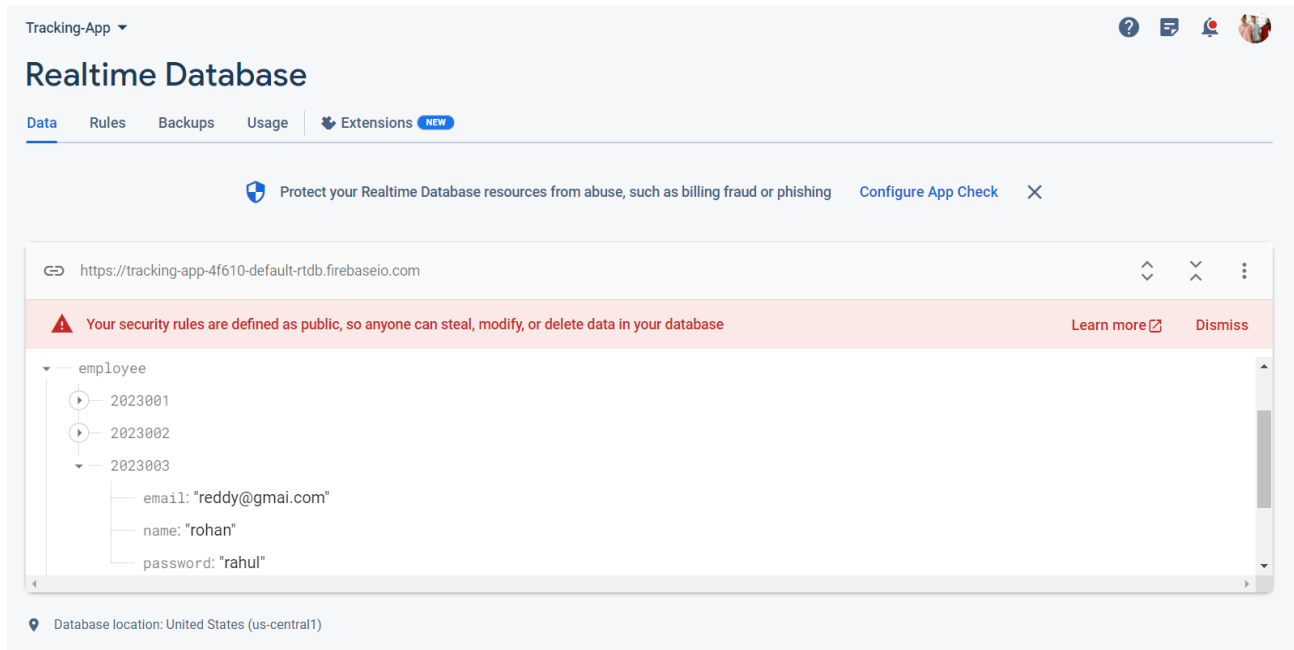


The screenshot shows the Firebase Realtime Database console. The URL is <https://tracking-app-4f610-default-rtbd.firebaseio.com/>. The database structure is as follows:

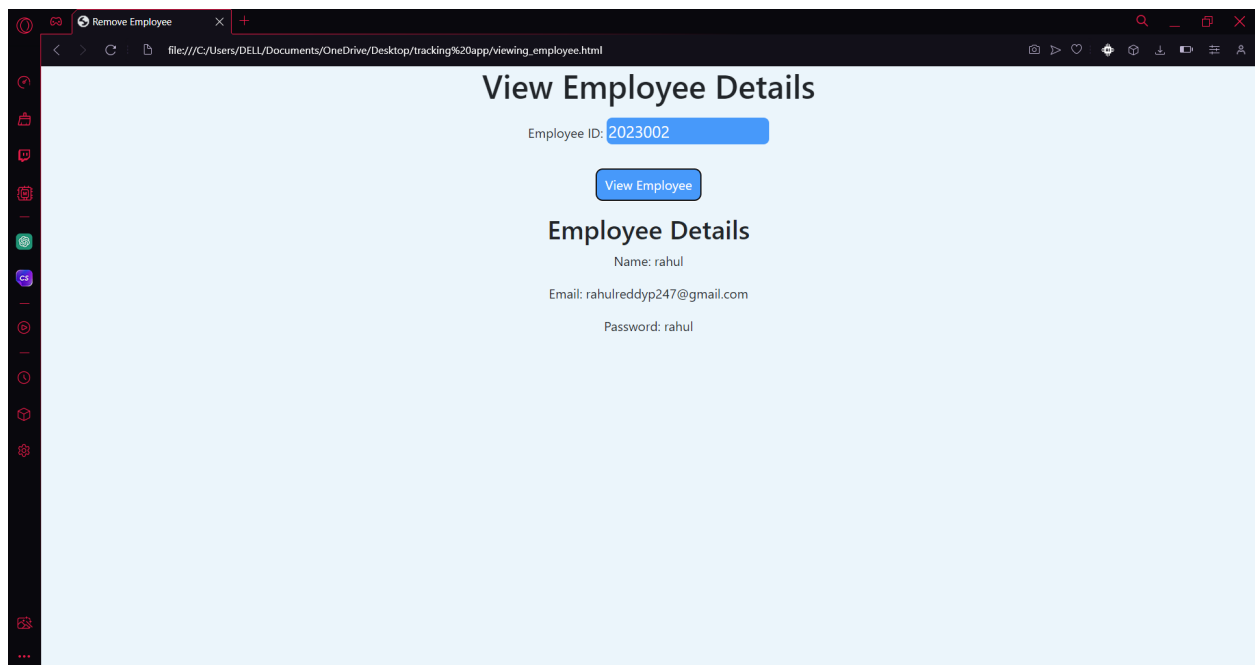
- Tasks
- employee
 - 2023001
 - 2023002
- locations

A warning message at the top states: "Your security rules are defined as public, so anyone can steal, modify, or delete data in your database".

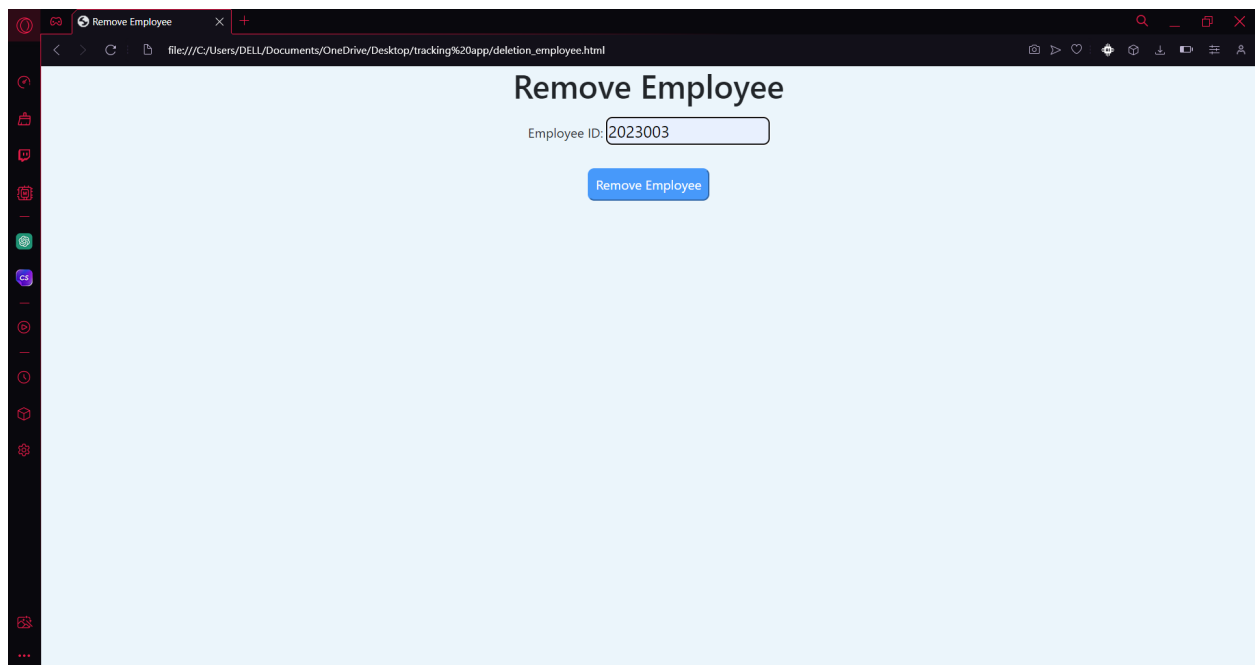
In database after adding employee in database:-



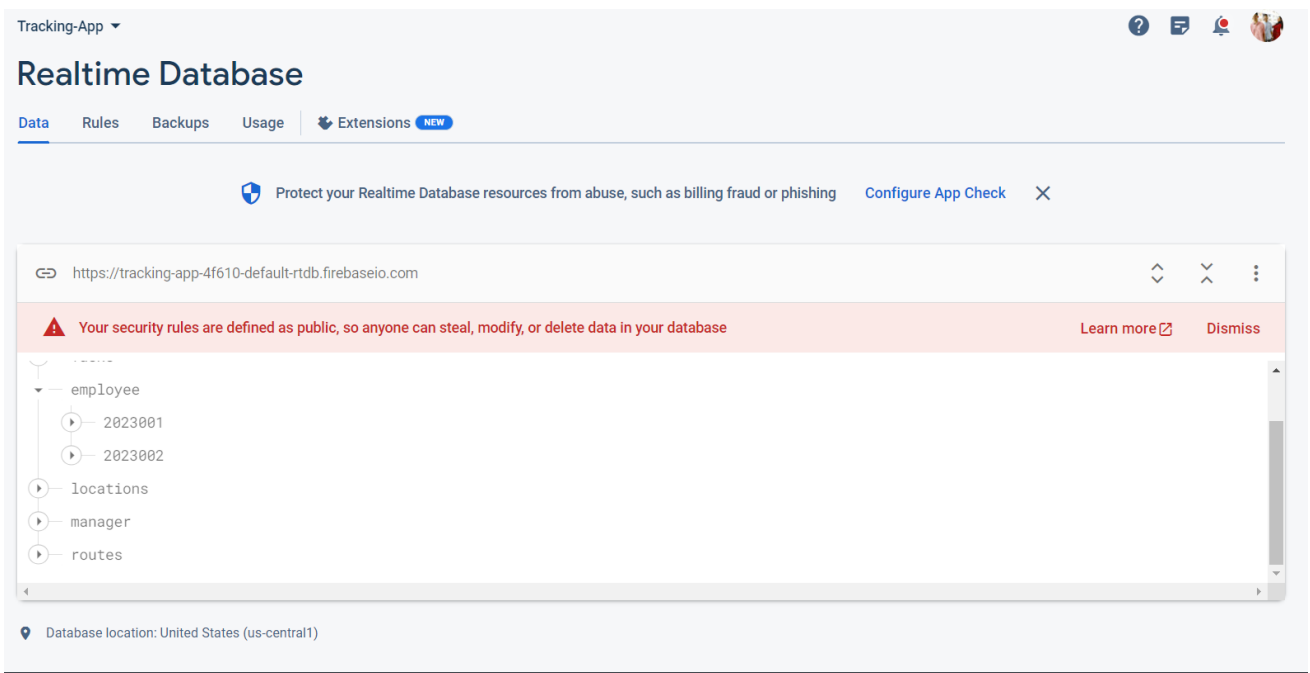
When we press Viewing Employee button:-



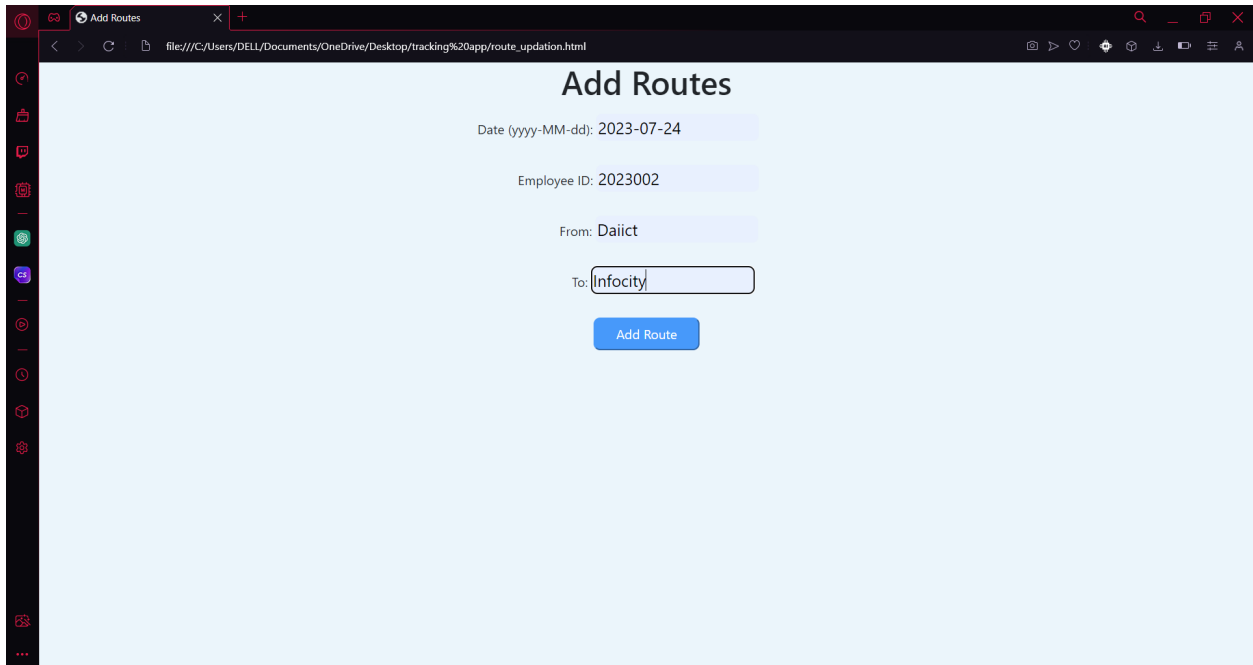
When we press remove employee button:-



We can see in the below with employee id of 2023003 is deleted which we added before

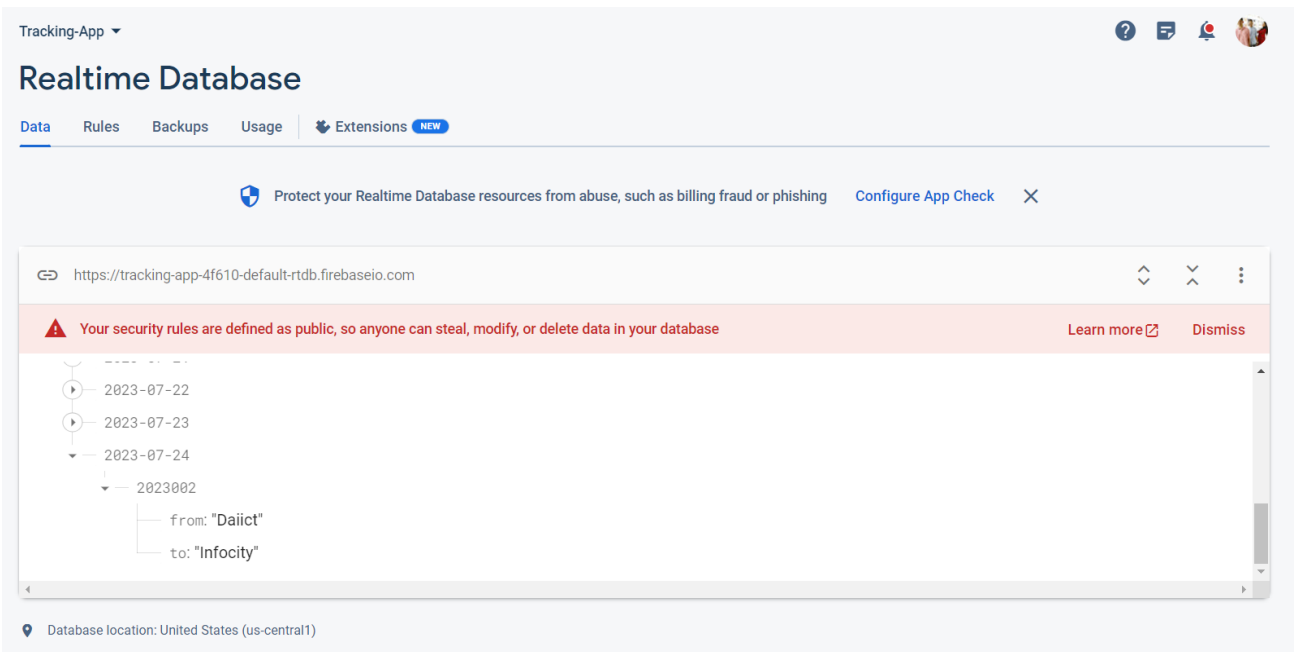


When we press Route management button:-



The screenshot shows a web browser window with the title 'Add Routes'. The address bar displays the file path: `file:///C:/Users/DELL/Documents/OneDrive/Desktop/tracking%20app/route_update.html`. The form contains the following fields and controls:

- Date (yyyy-MM-dd):** 2023-07-24
- Employee ID:** 2023002
- From:** Daiict
- To:** Infocity
- Add Route** button

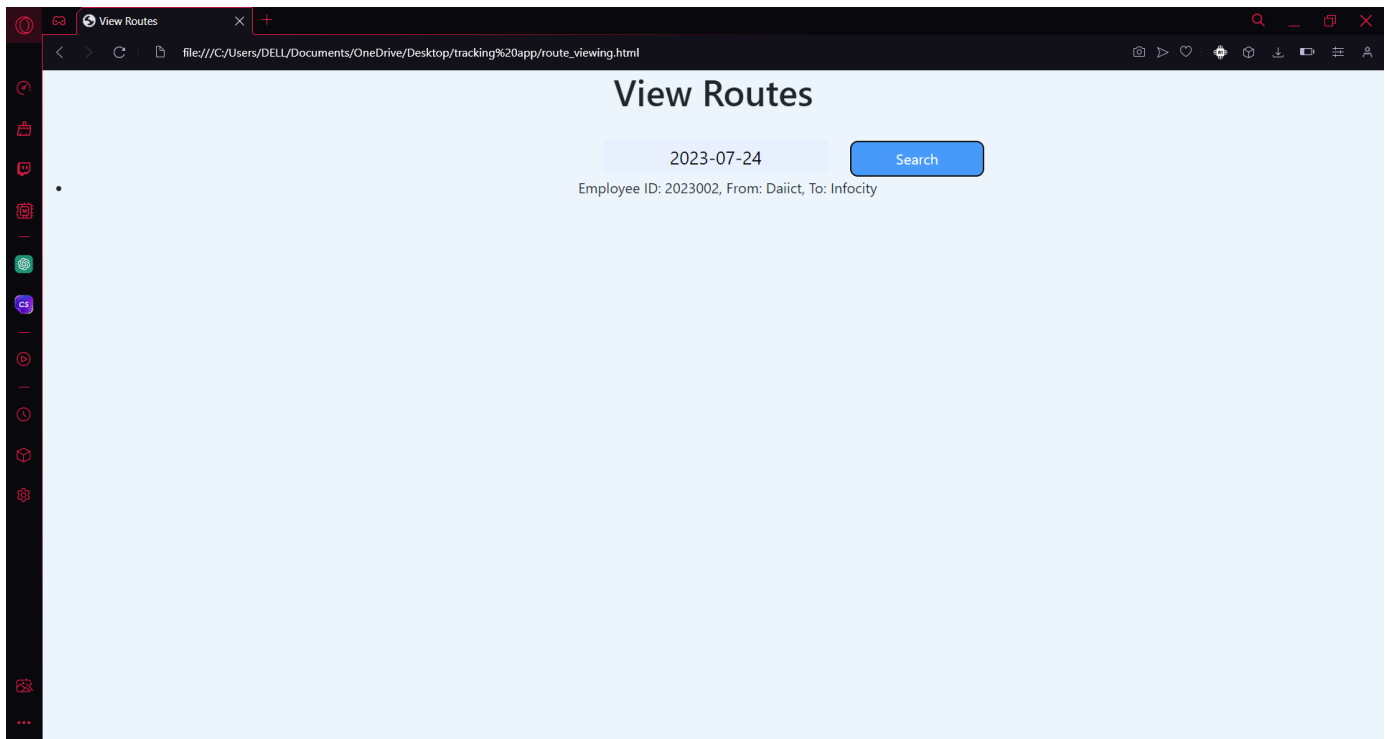


The screenshot shows the Firebase Realtime Database console for the 'Tracking-App'. The breadcrumb navigation is 'Tracking-App > Realtime Database'. The 'Data' tab is selected, showing a tree structure of the database contents:

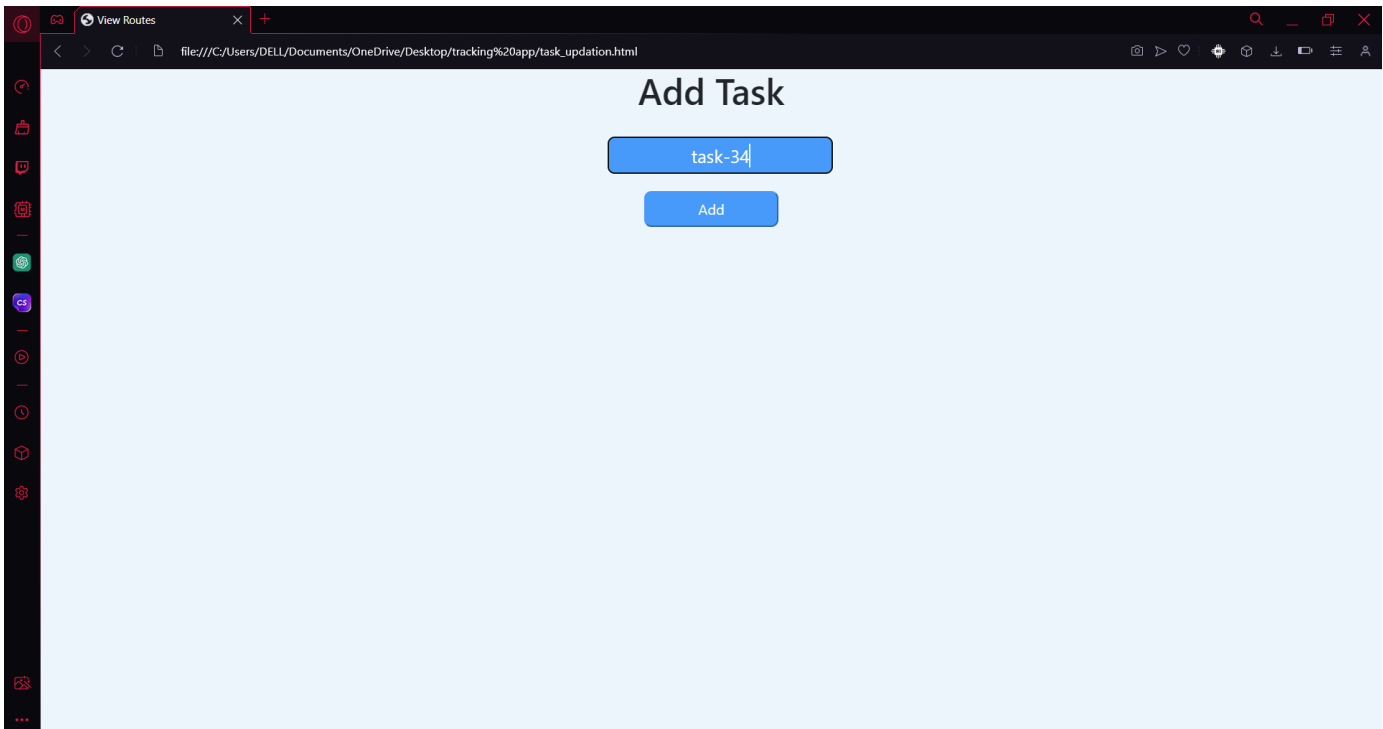
- 2023-07-22
- 2023-07-23
- 2023-07-24
 - 2023002
 - from: "Daiict"
 - to: "Infocity"

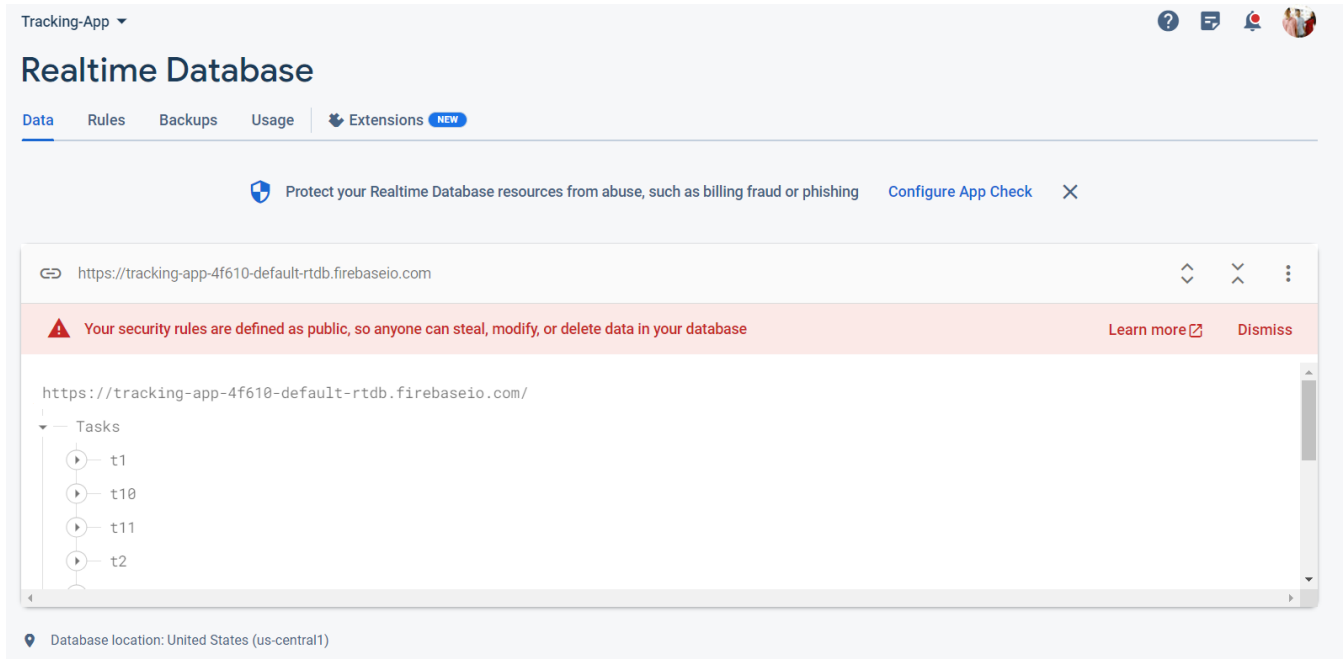
A security warning is displayed at the top: 'Your security rules are defined as public, so anyone can steal, modify, or delete data in your database'. The database location is noted as 'United States (us-central1)'.

When we route search button:-

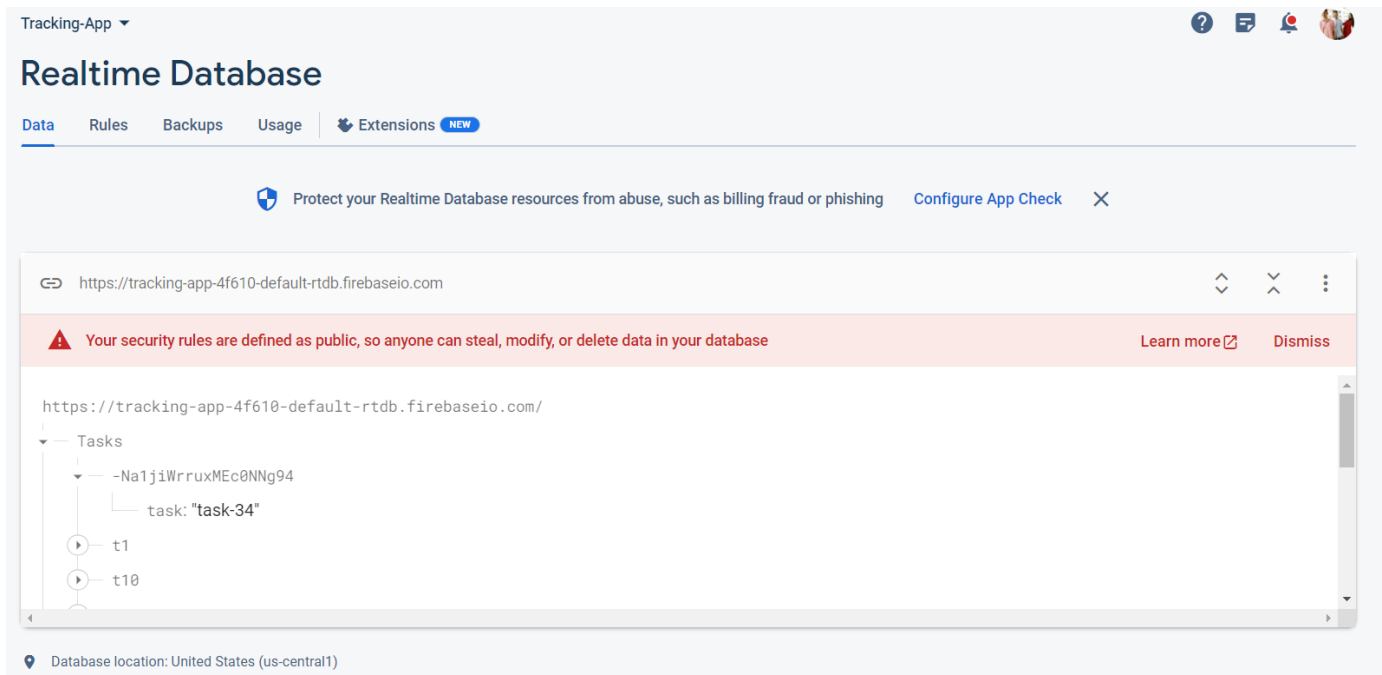


When we press Task Updation button:-



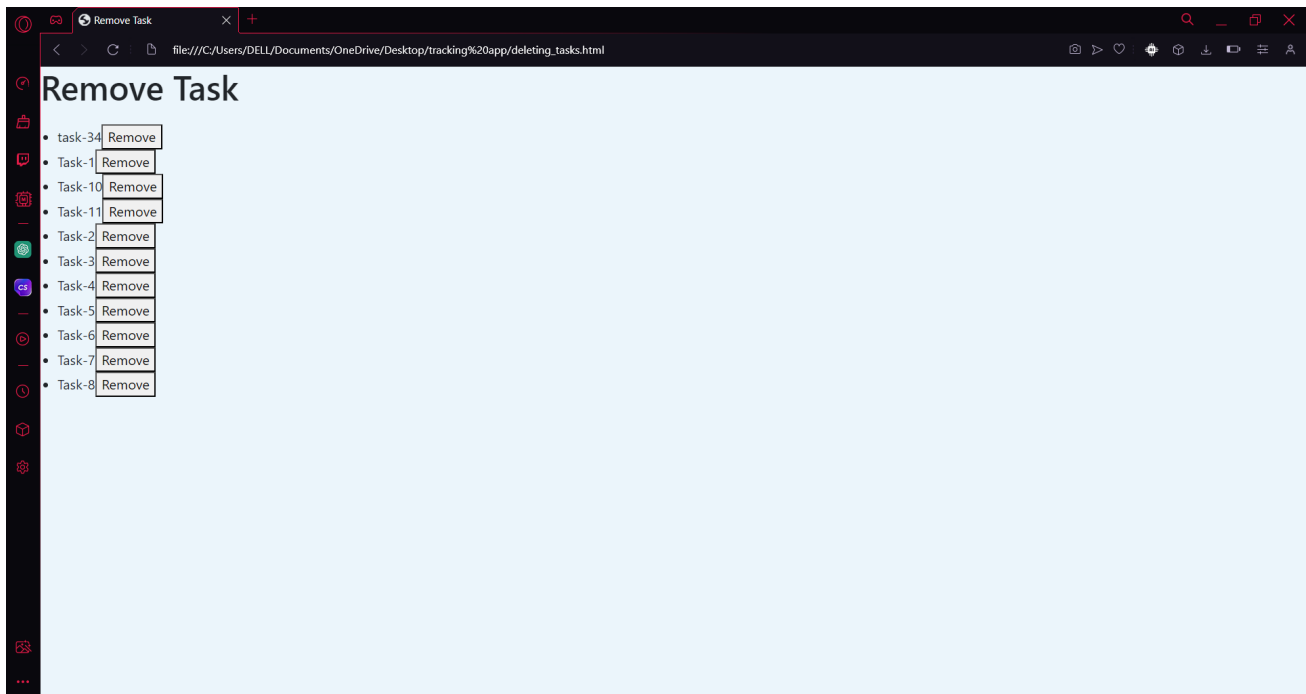


Above picture is the database before adding tasks

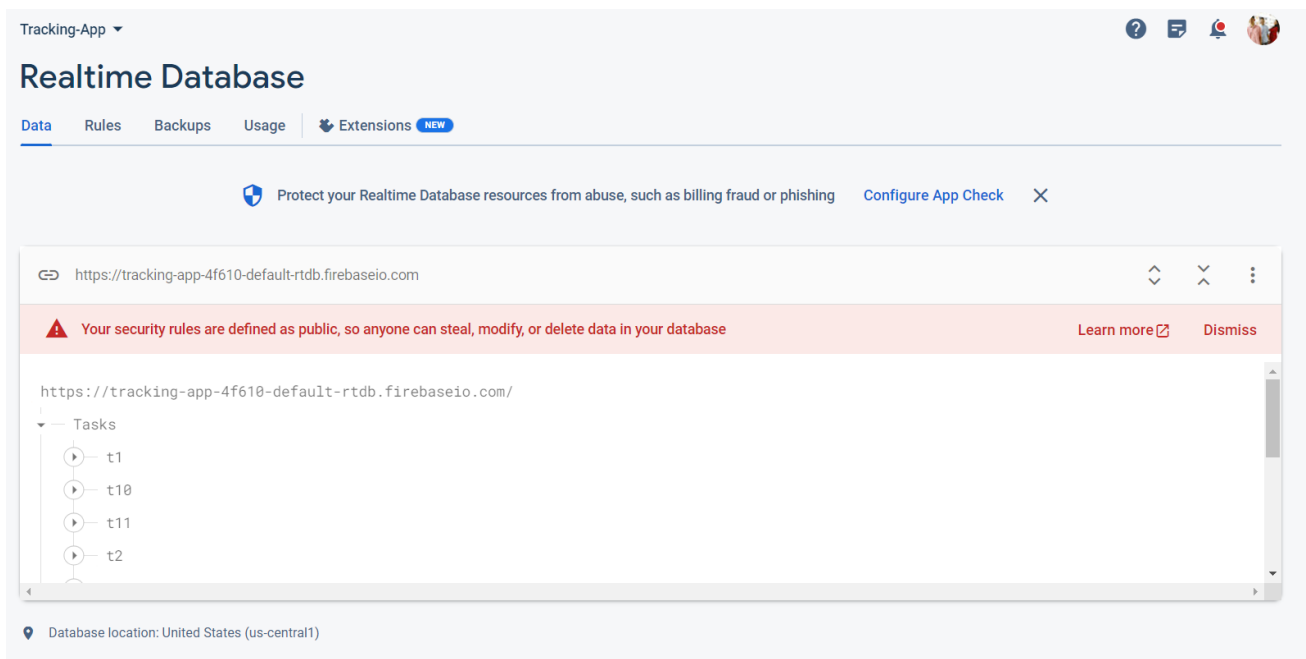


Above Picture is after adding the task

When we press Task deletion button:-

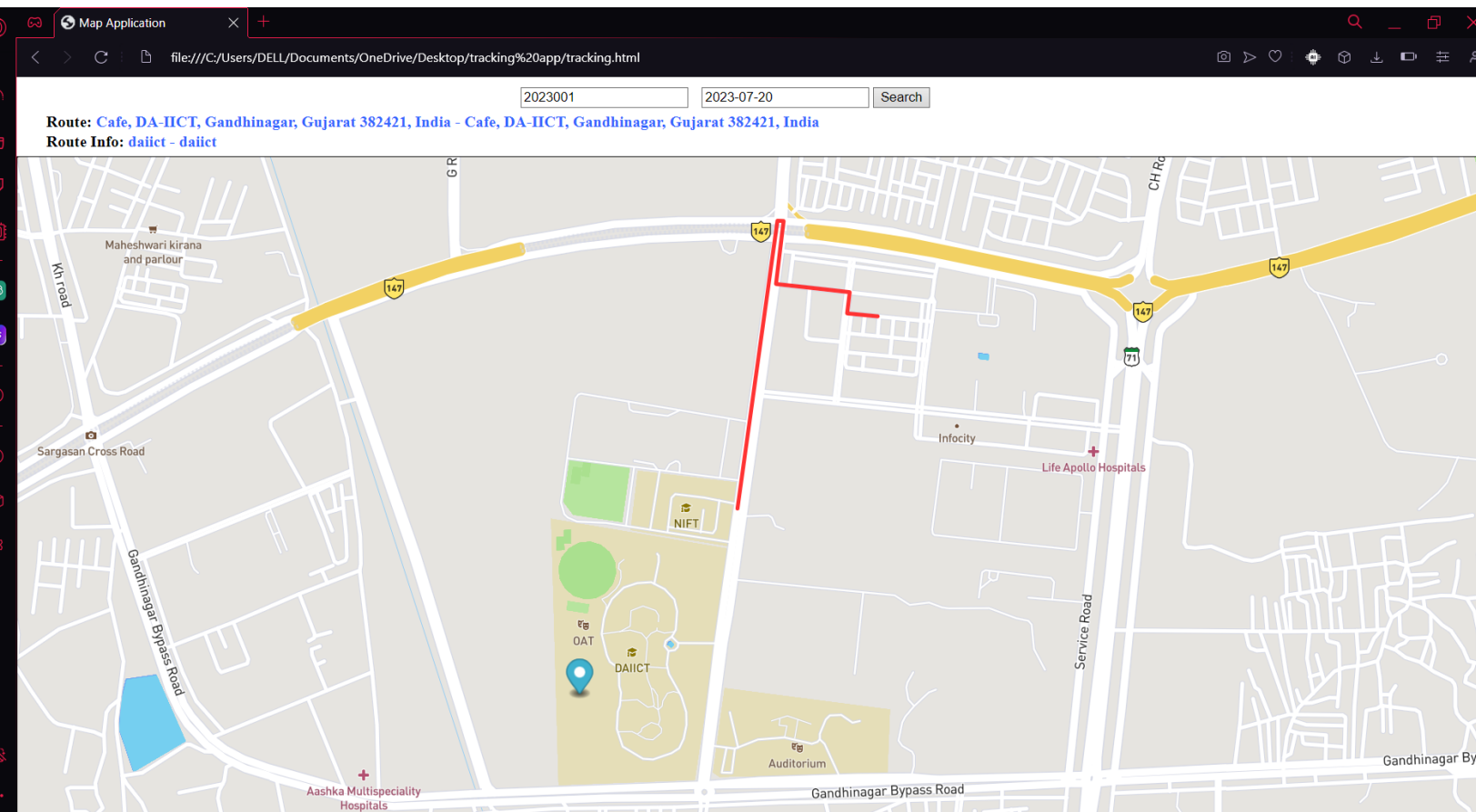


After deleting the task:-



Here we can see we deleted the task-34 task which we added above

When we press tracking button:-



- Here we have to search based on employee id and date.
- Here red line represents the route that the employee must go i.e the fiber line and marker represents the employee
- Here the route is the from and to location which should be done by the employee. Route info is the from and to location which is done by the employee.

Summary

The Pilot project is completed by creating the Mobile App for Patrollers and Web app for Managers using the above solution architecture. The mobile App and Web app have been demonstrated for successful usage for a maximum of 200 users and 50 kms.

This app can be further extended as a workforce management app by upgrading the technology and using the relevant Hardware.