

A MAJOR PROJECT REPORT ON

Road Management System

A report Submitted in partial fulfilment of the requirements of the degree

Bachelor of Technology IN COMPUTER SCIENCE AND ENGINEERING

Submitted by

P.CHARITHA(R170348)



Under the guidance of

**Ms. E.Susmitha
Assistant Professor**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES,

RK Valley,

AS A PART OF MAJOR PROJECT

May 2023.

Acknowledgement

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts success.

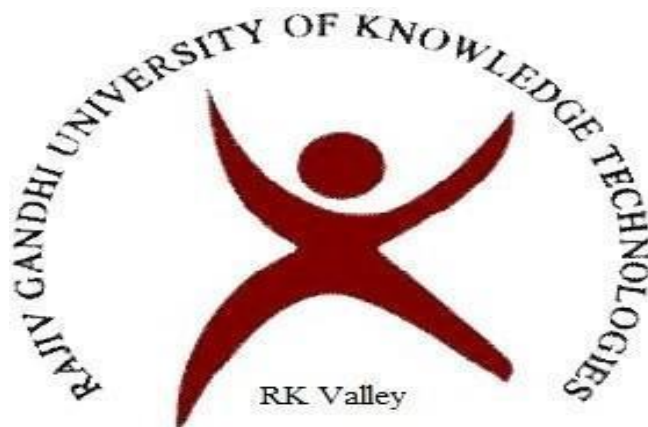
I am extremely grateful to our respected Director, Prof. K. SANDHYA RANI for fostering an excellent academic climate in our institution.

I also express my sincere gratitude to our respected Head of the Department Mr.N. Satyanandaram for his encouragement, overall guidance in viewing this project a good asset and effort in bringing out this project.

I would like to convey thanks to our guide at college Miss.E SUSMITHA for her guidance, encouragement, co-operation and kindness during the entire duration of the course and academics.

My sincere thanks to all the members who helped me directly and indirectly in the completion of project work. I express my profound gratitude to all our friends and family members for their encouragement.

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
RAJIV GANDHI UNIVERSITY OF KNOWLEDGE
TECHNOLOGIES ,RK VALLEY ,
May 2023.**



CERTIFICATE

This is to certify that the report entitled “**Road Management System**” submitted by P.Charitha(R170348) partial fulfilment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out by them under my supervision and guidance.

The report has not been submitted previously in part or in full to this or any other University or Institution for the award of any degree or diploma.

E.Susmitha,
Project Internal Guide,
Computer Science and Engineering,
R.K Valley, RGUKT.

N.Satyanandaram,
Head of the Department,
Computer Science and Engineering,
R.K.Valley, RGUKT.

DECLARATION

We are certifying that, I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature of the student

P.CHARITHA

R170348

ABSTRACT

The goal of this paper is to find out what are the steps in developing an application, the difficulty of it and future improvements. With this occasion, I had worked on the road maintenance known as ROAD AID. Roads are the transport infrastructures which play an important role in supporting the economic growth of a region. Therefore, it is necessary to do maintenance efforts so that the roads can function optimally. Recently, there is a rapid development of road transportation network. This situation arises due to the number of vehicles on the road that keeps increasing year by year. Thus, this will increase the possibility of dangerous situations to the road users if roads are not being maintained appropriately. Therefore, in order to keep the road in safe condition, the road management activity should be improved. So, Here in Cube highways provides a complete software system for road management. It invests in road and highway projects along with other select infrastructure sectors in India. Maintenance all activities such as accidents, issues, bills, safety, Assessment of road pavement conditions in the roads which are maintained by the company.

This project applications are user friendly and serviceable to the public. This project contains frontend, backend, and mobile application parts. I had worked on the backend development. We the cube members built a REST API as a layer over the ERP – the system used by the client so far. Frontend apps communicate with the ERP through the REST API in a much more simple and transparent way. This enabled user to speed and safety maintenance of the road.

Table of Contents

ACKNOWLEDGEMENT	1
CERTIFICATE	2
DECLARATION	3
ABSTRACT	4
1.INTRODUCTION	5
1.1 Purpose	
1.2 Features	
2.REQUIREMNT ANALYSIS	6
2.1 Requirement Specification.	
2.1.1 Functional Requirements.	
2.1.2 Hardware Requirements.	
2.1.3 Software Requirements.	
2.1.4 Technologies used.	
3.SOFTWARE ENVIRONMENT	8
4.IMPLEMENTATION	11
4.1 Graphical user interface.	
4.2 sample code.	
4.3 sample screenshots.	
5.Testing	25
5.1 Unit Testing.	
5.2 Integration Testing.	
6.Conclusion and Future scope.	26
6.1 Conclusion.	
6.2 Future scope	

1.Introduction

Road network is the most important infrastructure for a neighbourhood as it acts like the veins of a planned neighbourhood. Supervision and maintenance of these critical transportation assets can also be beneficial to obtain higher safety, convenience and cost effectiveness in the transportation sector for citizens. Road Management System (RMS) is a device or systematic approach that can furnish an inclusive inventory for road network and set up the work with the assist of much less time and struggle. The system can consider the roads and discover out a suitable maintenance desires with priorities underneath the accessible funds.

IT is a non-profit organization providing road maintenance services to the public. The organization was incepted with the objective of delivering comprehensive, speedy, reliable and quality services. There are apps and websites related to these services and our cube highways developed a website and ROAD AID app. Those apps have great features and functionalities to satisfy users.

Purpose and Features

The purpose and features of Road Maintenance System

- It prolongs the road life and postpones the day when renewal will be required.
- It reduces the cost of operating vehicles on roads.
- It helps to keep roads open and enables greater regularity, punctuality and safety of road transport services.
- Inspection date pavement conditions are updated and tracked over time.

Our road asset management system software allows all pavement-related work to be captured on work orders from small patch and repair jobs to complete structural improvements. Using the road maintenance app and website ,work orders have the ability to capture the total of the work(i.e. labour hours ,equipment hours, and materials), whether the work is done in-house or outsourced. Work order pavement categories such as preventative repair, rehab, or structural improvement can be defined.

2. Requirement ANALYSIS

2.1 Requirement Specification

Simple and interactive GUI: System facilitates simple and interactive Graphical User Interface for the user while handling the system. Android is an open sourcing mobile operating system based on Linux which is a completely open and integrated platform for mobile devices.

Database to store and retrieve the data. Functional requirement of the system describes what the system does. The main functional requirements of this system are as follows:

hardware Requirements

Processor: I3

Har disk: 64GB

RAM: 4GB

2.1.3 Software Requirements.

Android OS: From Android 7

Android Studio: 3

2.2 Technologies used:

Node JS:

Goal - To understand the basics of backend development using node js and to host a server using express js.

Explanation – Node.js is an open-source, cross-stage, JavaScript runtime condition that executes JavaScript code outside of an internet browser. Node.js lets engineers use JavaScript to compose order line devices and for server-side scripting running contents server-side to deliver dynamic website page content before the page is sent to the client's internet browser. Subsequently, Node.js speaks to a "JavaScript all over the place" worldview, bringing together web-application improvement around a solitary programming language, as opposed to various dialects for server-and customer side contents. Node.js has an occasion driven design equipped for offbeat I/O. These plan decisions mean to upgrade throughput and adaptability in web applications with many info/yield tasks, just as for continuous Web applications (e.g.,

ongoing correspondence projects and program games). The Node.js disseminated advancement venture was recently represented by the Node.js Foundation, and has now converged with the JS Foundation to frame the OpenJS Foundation, which is encouraged by the Linux Foundation's Collaborative Projects program.

Express JS:

Express.js is the most popular backend framework for Node.js, and it is an extensive part of the JavaScript ecosystem. It is designed to build single-page, multi-page, and hybrid web applications, it has also become the standard for developing backend applications with Node.

Mongo DB:

MongoDB is a cross-platform document-oriented database program. It provides Open-source NoSQL data storage for modern applications and is the most popular non-relational database in the world. MongoDB has many advantages, such as scalability and performance, which make it an ideal choice for backend developers. It gives backend developers many benefits that may be difficult to find anywhere else for developers. It provides various services and tools to integrate with the database, such as Atlas Search, MongoDB Charts, Performance Advisor, MongoDB Compass and multi-cloud deployment.

3. SOFTWARE ENVIRONMENT

Mongo DB:

MongoDB is a powerful and highly scalable database, suitable for use on any application's backend.

- Allows you to easily scale out your database horizontally using sharding, which distributes your data across multiple servers for better performance and availability.
- With MongoDB's flexible schema, developers can also move their data around, modify the structure of their collections, and update the server with minimal effort.
- MongoDB provides extensive analytical capabilities that enable fast and intuitive aggregation, reporting, and analytics applications suitable for real-time analytics. From the MongoDB Compass GUI to the in-browser shell, MongoDB offers a variety of help-full tools to help developers work with their data more efficiently. Here are some MongoDB Developer Tools:

MongoDB Developer Tools:

From the MongoDB Compass GUI to the in-browser shell, MongoDB offers a variety of help-full tools to help developers work with their data more efficiently. Here are some MongoDB Developer Tools:

- **The Mongo Shell/Mongo Script Shell**

The Mongo Shell is an interactive command-line script shell that allows you to manage and query your databases easily. It also supports the JavaScript API for interactive database operations, query automation, and administration.

- **The GUI Client: MongoDB Compass**

MongoDB Compass is a GUI client that provides detailed visibility into your database schema (including table creation, indexing, and document validation). It also provides support for ad-hoc queries & visualizations of document collections.

- **Database Explorer & Query Builder Tool**

The Database Explorer tool makes it easy to explore your database collections and dig deep-er into combined documents & sub-records. This helps developers quickly debug issues and find answers to complex datasets easily.

MongoDB Developer Tools Offers:

MongoDB offers a wide range of use cases for backend development. Its unique capabilities allow it to process, analyse, and aggregate data like never before.

- **Data Aggregation:** MongoDB's aggregation pipeline enables developers to process large sets of data quickly, efficiently, and effectively.

- **Full-Text Search:** MongoDB's text indexing feature allows you to search database documents with minimal effort.
- **Security:** MongoDB Developer Tools take the guesswork out of data security. Whether it's managing user roles and privileges, or setting up SSL/TLS encryption protocols, MongoDB provides the infrastructure you need to keep your data safe.

POSTMAN:

POSTMAN is an API client used to develop, test, share and document APIs. It is used for backend testing where we enter the end-point URL, it sends the request to the server and receives the response back from the server. The same thing can be accomplished through API Templates like Swagger as well.

Reduce time to production

Postman mocks support split-stack development so front-end and back-end developers can work in parallel and view responses without spinning up a back end.

Communicate your API's expected behavior

View API responses with the applications and services waiting to use it before and during development.

Catch and fix bugs early

Get better visibility into errors so you can debug more quickly.

4. IMPLEMENTATION

4.1 Graphical user interface

The user interface is kept simple and understandable. The user need not take any additional effort to understand the functionality and navigation in the application. The UI designing should be easily understandable and should know where the input is given. Hints are given to help the user in giving the correct input.

4.2 Sample Code & 4.3 Screenshots

1. Developed a new custom route for IMS user analytics.

In this I have written a code for getting user details based on the requirements and conditions they are provided. I have used postman tool for checking the output of the code. When you enter the URL of developed route in the post man by selecting get method the output will be generated for the user analysis. The output data used in frontend to show the analysis in website.

Code :

```
export const getImsAnalytics = async (req: Request, res: Response, next:
NextFunction) => {
  const road = cls.get('road');
  if (!req.query.CRO && !req.query.RPO && !req.query.Paramedical
&& !req.query['Incident Manager'] &&
!req.query['Hydra Operator'] && !req.query.Helper) {
    const errorObj: ErrorObj = {
      message: 'Please enter parameters',
      code: 1006,
      status: httpCodes.FORBIDDEN
    };
    const error = new HAMSError(errorObj);
    logger.error(error.message);
    return next(error);
  }
  let imsUser: any = {} ;
  if (req.query.CRO) {
    imsUser = { $or: [{ imsCRO : req.query.CRO }, { additionalStaff :
req.query.CRO } ] };
  } else if (req.query.RPO) {
    imsUser = { $or: [{ imsRPO : req.query.RPO }, { additionalStaff :
req.query.RPO } ] };
  } else if (req.query.Helper) {
```

```

    imsUser = {$or: [{ imsHelper : req.query.Helper }, { additionalStaff :
req.query.Helper }]}];
  } else if (req.query.Paramedical) {
    imsUser = {$or: [{ imsRPO : req.query.Paramedical }, { additionalStaff :
req.query.Paramedical }]}];
  } else if (req.query['Hydra Operator']) {
    imsUser = {$or: [{ imsRPO : req.query['Hydra Operator'] }, { additional-Staff :
req.query['Hydra Operator'] }]}];
  } else if (req.query['Incident Manager']) {
    imsUser = {$or: [{ imsRPO : req.query['Incident Manager']},
    { additionalStaff : req.query['Incident Manager']}, { imsHelper :
req.query['Incident Manager']},
    { imsCRO: req.query['Incident Manager']}]}}];
  }
  if ((req.query.dateFrom && !req.query.dateTo) || (req.query.dateTo
&& !req.query.dateFrom)) {
    const errorObj: ErrorObj = {
      message: 'MISSING - From and To Dates',
      code: 1006,
      status: httpCodes.FORBIDDEN
    };
    const error = new HAMSError(errorObj);
    logger.error(error.message);
    return next(error);
  }
  let query: any = {};
  if (req.query.dateFrom && req.query.dateTo) {
    if (new Date(req.query.dateFrom) <= new Date(req.query.dateTo)) {
      query = {$and : [imsUser, {roadId : road._id} ,
      {createdAt : {$gte : new Date(req.query.dateFrom) , $lte : new
Date(req.query.dateTo)}}]}];
    } else {
      const errorObj: ErrorObj = {
        message: 'To Date should be greater than From Date',
        code: 1007,
        status: httpCodes.FORBIDDEN
      };
      const error = new HAMSError(errorObj);
      logger.error(error.message);
      return next(error);
    }
  } else {
    query = { $and : [imsUser, {roadId : road._id}]}];
  }
}

```

```

const shiftReports: any = await ShiftReport.find(query).select({startHour: 1 ,
endHour: 1, startMinute: 1,
endMinute: 1 , createdAt: 1, roadAidUserNameId: 1}).lean();
const imsAnalytics = {
  total : {
    shiftReports : 0,
    AA: 0,
    AR: 0,
    IR: 0,
    IA: {
      total: 0,
      deadAnimal: 0,
      unAttendVeh: 0,
      unWantedMat: 0,
      vehBreak: 0
    }
  },
  currentMonth : {
    shiftReports : 0,
    AA: 0,
    AR: 0,
    IR: 0,
    IA: {
      total: 0,
      deadAnimal: 0,
      unAttendVeh: 0,
      unWantedMat: 0,
      vehBreak: 0
    }
  },
  lastMonth : {
    shiftReports : 0,
    AA: 0,
    AR: 0,
    IR: 0,
    IA: {
      total: 0,
      deadAnimal: 0,
      unAttendVeh: 0,
      unWantedMat: 0,
      vehBreak: 0
    }
  }
};
if (shiftReports && shiftReports.length) {

```

```

const updateArr: any = [];
const roadAidUserIds: any = [];
let shiftStartDate: any ;
let shiftEndDate: any ;
const currentMonthStart = new Date(moment().startOf('month').format());
const lastMonthStart = new Date(moment().subtract(1,
'months').startOf('month').format());
const lastMonthEnd = new Date(moment().subtract(1,
'months').endOf('month').format());
for (const shift of shiftReports) {
    if (shift.startHour !== undefined && shift.endHour !== undefined &&
shift.startMinute !== undefined &&
    shift.endMinute !== undefined) {
        shiftStartDate = new
Date(shift.createdAt.setHours(Number(shift.startHour),
    Number(shift.startMinute), 0));
        shiftEndDate = new Date(shift.createdAt.setHours(Number(shift.endHour),
    Number(shift.endMinute), 0));
        imsAnalytics.total.shiftReports++;
        if (new Date(shift.createdAt) >= currentMonthStart) {
            imsAnalytics.currentMonth.shiftReports++;
        } else if (new Date(shift.createdAt) >= lastMonthStart && new
Date(shift.createdAt) <= lastMonthEnd) {
            imsAnalytics.lastMonth.shiftReports++;
        }
        if (shift.roadAidUserNameId &&
            !roadAidUserIds.some((id: any) => Ob-
jectID(id).equals(ObjectID(shift.roadAidUserNameId)))) {
            roadAidUserIds.push(ObjectID(shift.roadAidUserNameId));
        }
        updateArr.push({ createdAt: { $gte: shiftStartDate , $lte: shift-EndDate } });
    }
}
query.filter = { $or: updateArr , imsUnit: 'AMB', roadId: road._id };
query.select = { createdBy: 1, createdAt: 1 };
const accidents: any = await getAccIdentificationDb(query);
query.filter = { $or: updateArr, roadId: road._id };
query.select = { createdBy: 1, issueTypeId: 1, createdAt: 1, properties: 1 };
const issues: any = await getIssuesDb(query);
for (const acc of accidents) {
    let flag = 0;
    if (roadAidUserIds.some((id: any) => Ob-
jectID(id).equals(ObjectID(acc.createdBy)))) {
        imsAnalytics.total.AR++;
        flag = 1;
    }
}

```

```

    }
    if (new Date(acc.createdAt) >= currentMonthStart) {
        if (flag === 1) {
            imsAnalytics.currentMonth.AR++;
        }
        imsAnalytics.currentMonth.AA++;
    } else if (new Date(acc.createdAt) >= lastMonthStart && new
Date(acc.createdAt) <= lastMonthEnd) {
        if (flag === 1) {
            imsAnalytics.lastMonth.AR++;
        }
        imsAnalytics.lastMonth.AA++;
    }
    imsAnalytics.total.AA++;
}
const issueType = await IssueType.findOne({name: 'Incidents'}).select({_id:
1}).lean().exec();
for (const issue of issues) {
    if (roadAidUserIds.some((id: any) => Ob-
jectID(id).equals(ObjectID(issue.createdBy)))) {
        imsAnalytics.total.IR++;
        if ((issueType._id).equals(issue.issueTypeId)) {
            imsAnalytics.total.IA.total++;
            if (issue.properties['Dead Animal'] === true) {
                imsAnalytics.total.IA.deadAnimal++;
            }
            if (issue.properties['Unattended Vehicle'] === true) {
                imsAnalytics.total.IA.unAttendVeh++;
            }
            if (issue.properties['Unwanted Material'] === true) {
                imsAnalytics.total.IA.unWantedMat++;
            }
            if (issue.properties['Vehicle Breakdown'] === true) {
                imsAnalytics.total.IA.vehBreak++;
            }
        }
    }
}
if (new Date(issue.createdAt) >= currentMonthStart &&
roadAidUserIds.some((id: any) => Ob-
jectID(id).equals(ObjectID(issue.createdBy)))) {
    imsAnalytics.currentMonth.IR++;
    if ((issueType._id).equals(issue.issueTypeId)) {
        imsAnalytics.currentMonth.IA.total++;
        if (issue.properties['Dead Animal'] === true) {
            imsAnalytics.currentMonth.IA.deadAnimal++;
        }
    }
}

```



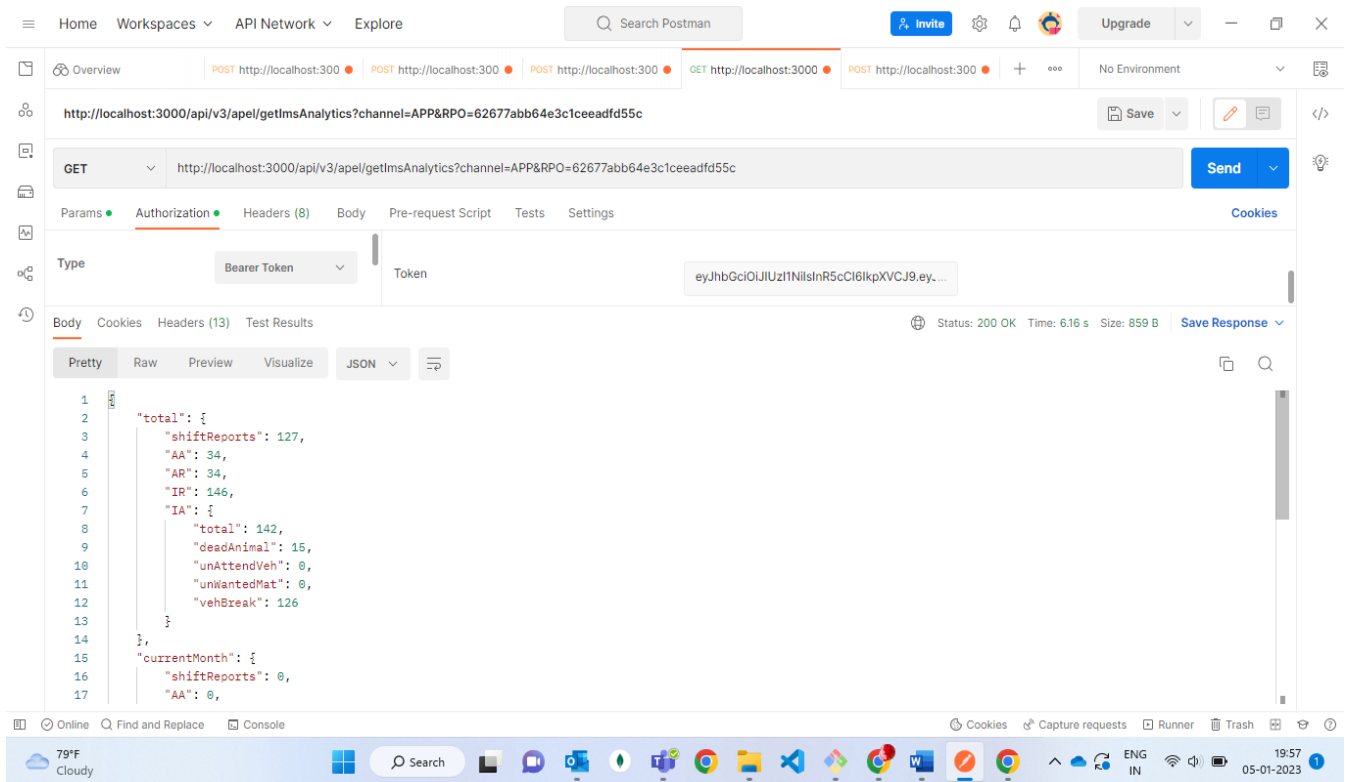
```

    }
    if (issue.properties['Unattended Vehicle'] === true) {
        imsAnalytics.currentMonth.IA.unAttendVeh++;
    }
    if (issue.properties['Unwanted Material'] === true) {
        imsAnalytics.currentMonth.IA.unWantedMat++;
    }
    if (issue.properties['Vehicle Breakdown'] === true) {
        imsAnalytics.currentMonth.IA.vehBreak++;
    }
}

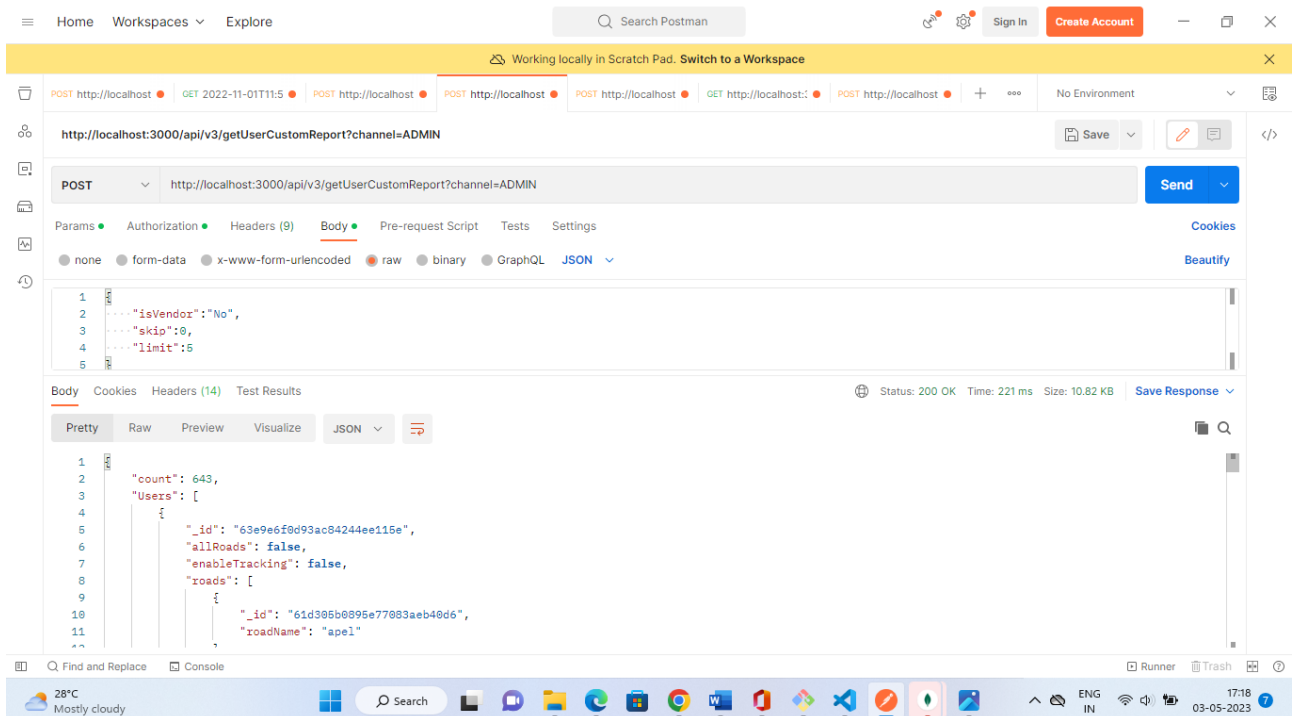
    } else if (new Date(issue.createdAt) >= lastMonthStart && new
Date(issue.createdAt) <= lastMonthEnd &&
    roadAidUserIds.some((id: any) => Ob-
jectID(id).equals(ObjectID(issue.createdBy)))) {
    imsAnalytics.lastMonth.IR++;
    if ((issueType._id).equals(issue.issueTypeId)) {
        imsAnalytics.lastMonth.IA.total++;
        if (issue.properties['Dead Animal'] === true) {
            imsAnalytics.lastMonth.IA.deadAnimal++;
        }
        if (issue.properties['Unattended Vehicle'] === true) {
            imsAnalytics.lastMonth.IA.unAttendVeh++;
        }
        if (issue.properties['Unwanted Material'] === true) {
            imsAnalytics.lastMonth.IA.unWantedMat++;
        }
        if (issue.properties['Vehicle Breakdown'] === true) {
            imsAnalytics.lastMonth.IA.vehBreak++;
        }
    }
}
    }
}
    }
return res.json(imsAnalytics);
};

```

OUTPUT:



2. Added new route for GetCustomUserReport



In this total users existed report data is returned by the function. From the frontend they give a call to the route using API calls then the data or json will be returned the

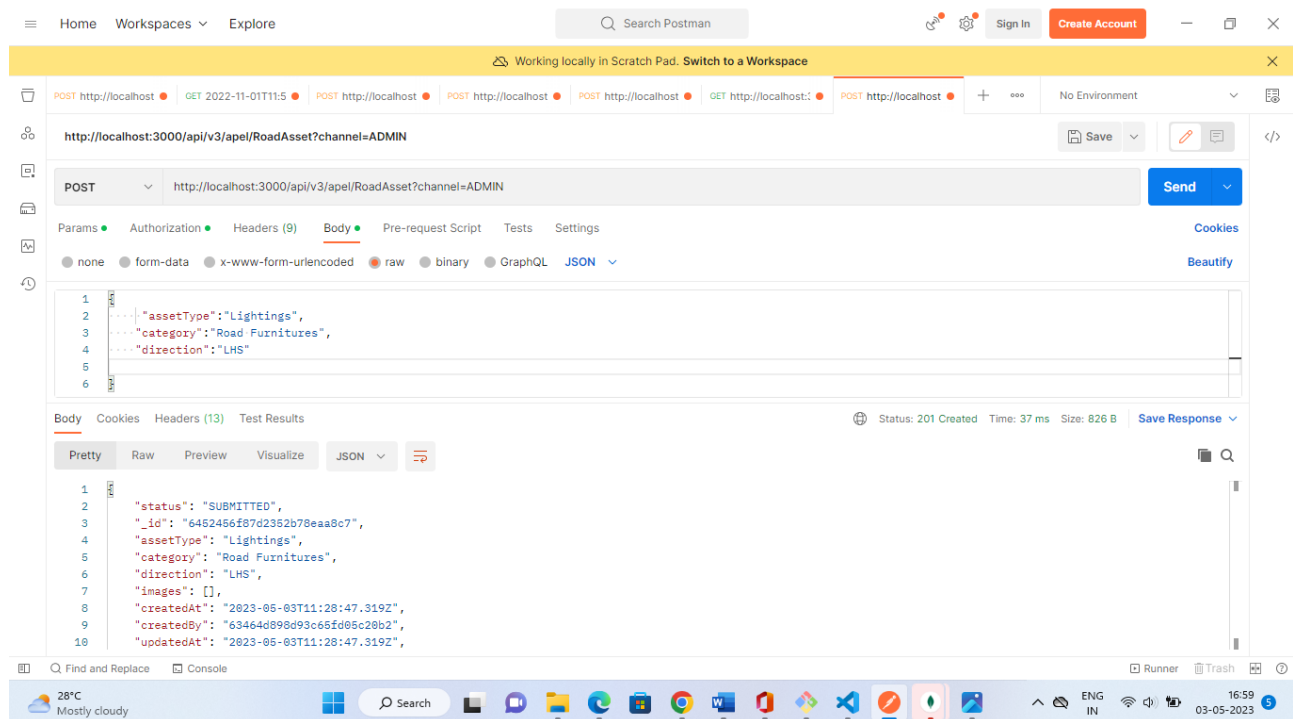
data is filtered based on their requirements from the backend the code is develop based on the requirements. All the fields related to the user based on the custom requirements we developed the page and functionality. This postman is esting purpose only from the backend. After integrating from frontend or app the data will be shown on the app or page.

3. Added new database called RoadAssets and History

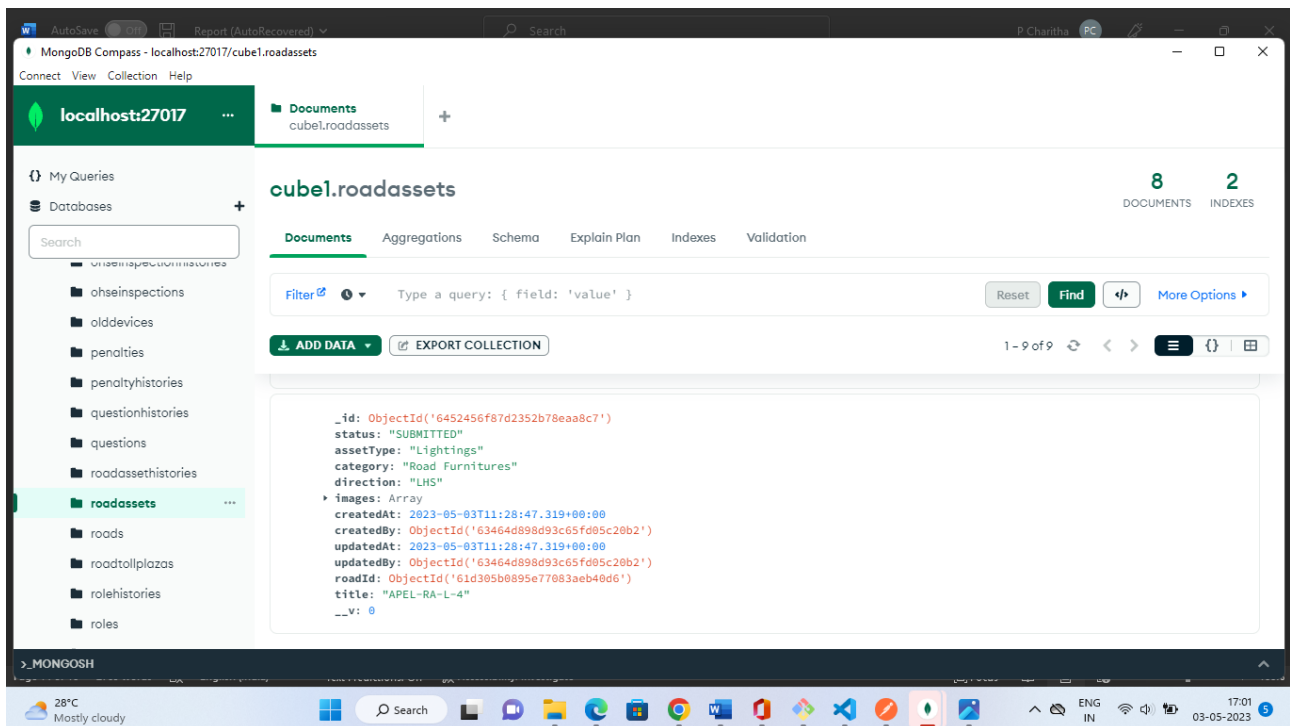
In this when user entered data related to road assets page ,the total data entered will be stored in road assets schema. New Schema created for storing and retrieving the data and the fields related to the road assets will be added in the schema model creation.

RoadAsset model code:

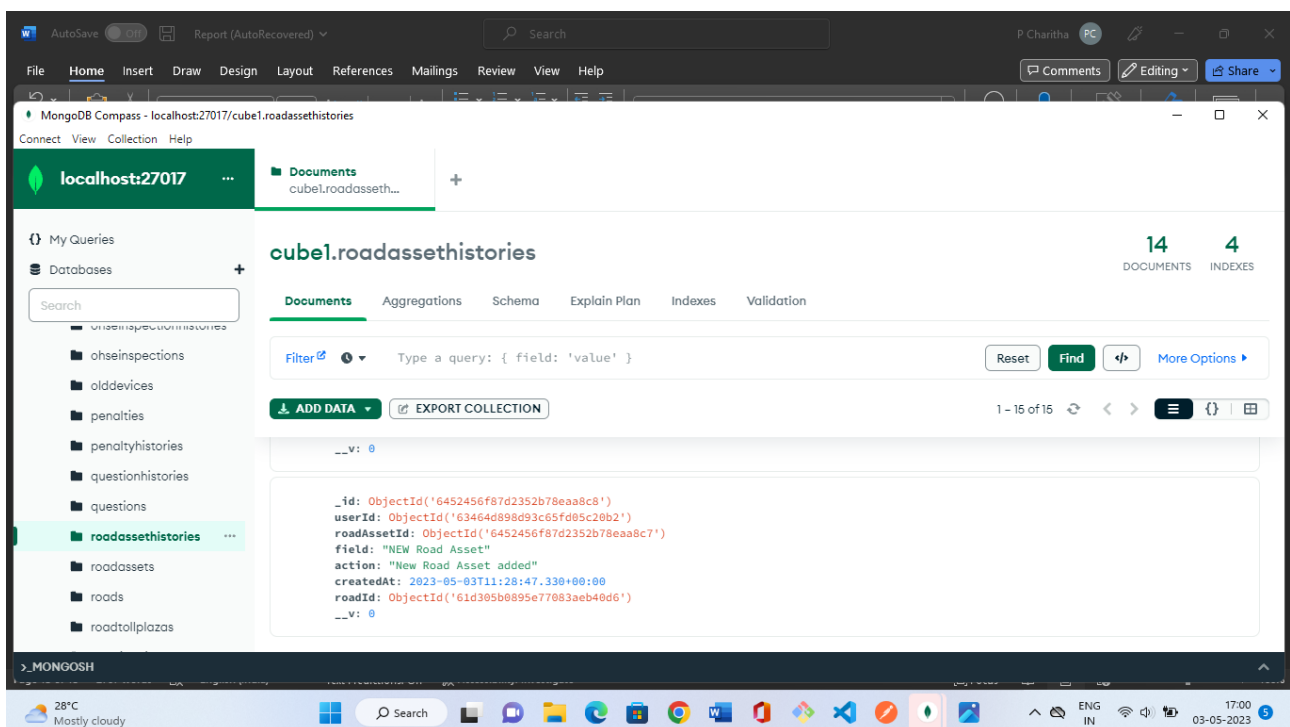
Tested using postman:



Data stored in database called roadAssets:



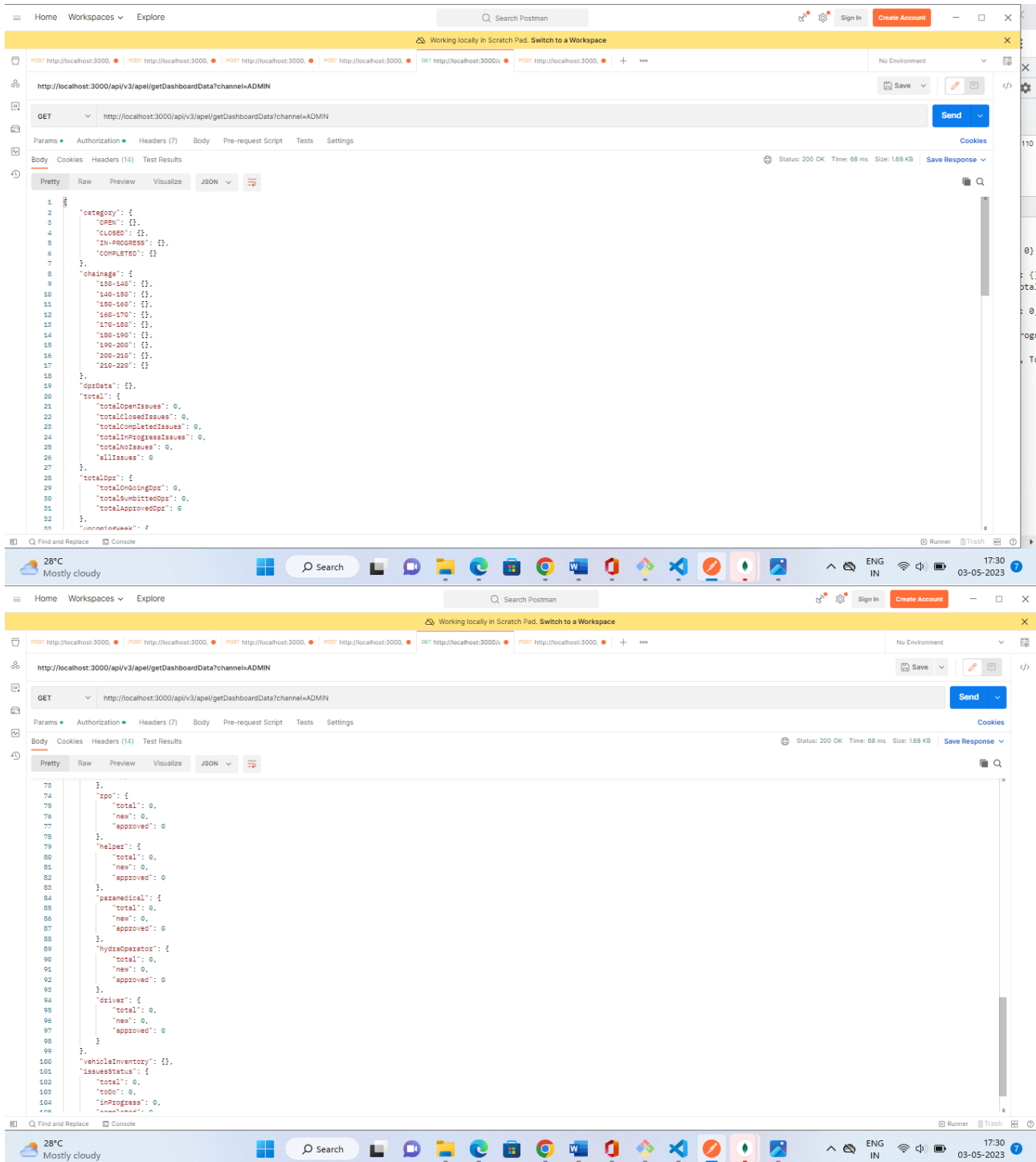
Along with road assets schema creation when any changes in existing assets or new assets are added then the changes will be added in roadassethistory schema.



4.added changes to get dashboard data

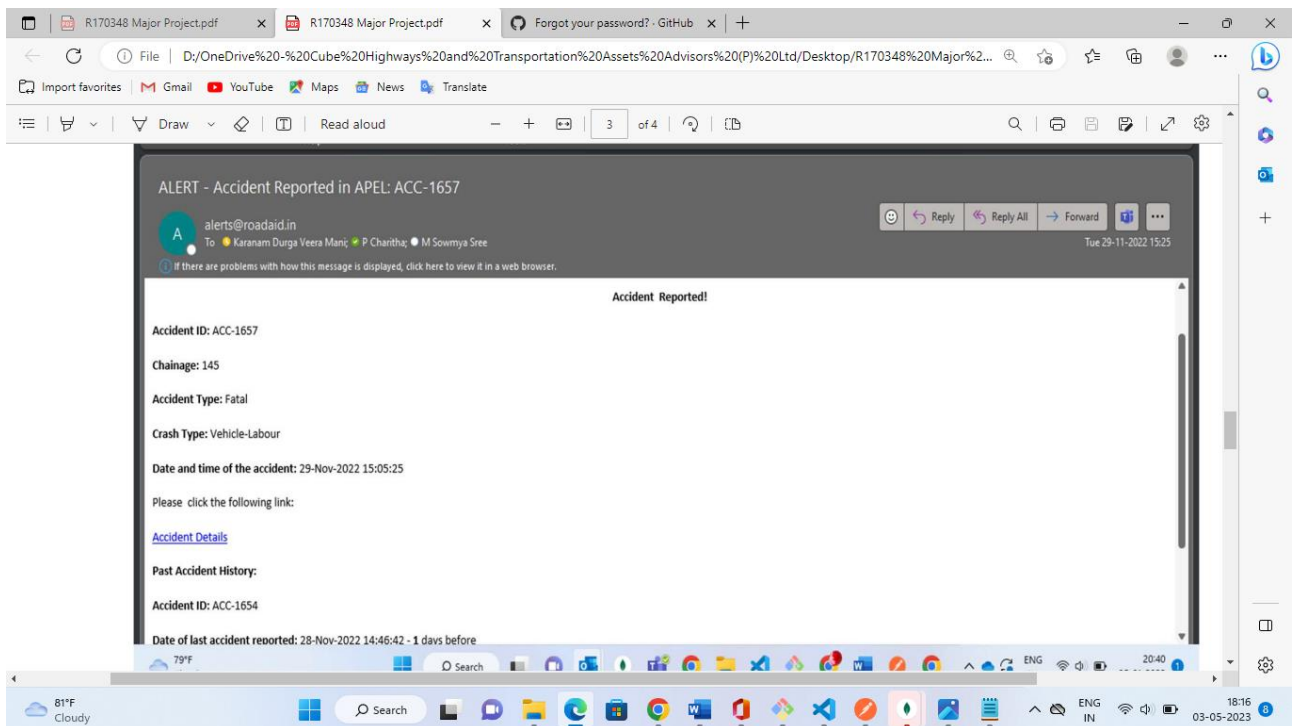
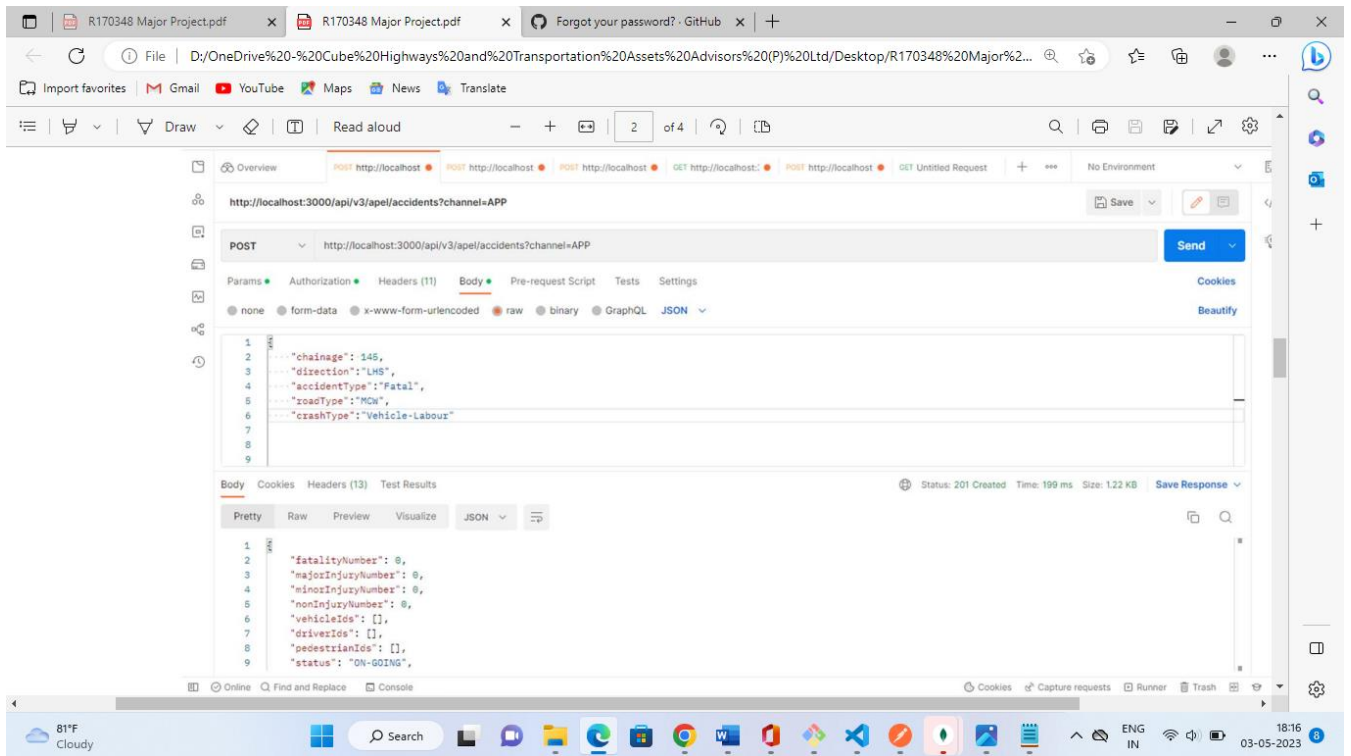
In this when we call the getdashboarddata route the data will be returned related accidents, ims users, chainages, dpr data, vehicle inventory and issues related

total data based on the customer requirements the data is calculated and it is returned from the backend when the route is called from frontend or app.



5. Developed a code for adding accident details in accident report main while creating new accident

In this when the URL entered the mail will be generated and sent to the users based on the conditions and the requirements.



6. Added validation for expected completion date while assigning the issue to user

When the expected completion date is less than today then it shows a logger message with error.

message with error

R170348 Major Project.pdf x R170348 Major Project.pdf x Forgot your password? - GitHub x +

File D:/OneDrive%20-%20Cube%20Highways%20and%20Transportation%20Assets%20Advisors%20(P)%20Ltd/Desktop/R170348%20Major%20... Import favorites Gmail YouTube Maps News Translate

Draw Read aloud 3 of 4

Home Workspaces API Network Explore Search Postman

Overview POST http://localhost:3000/api/v3/apel/assignUserToIncident?channel=APP No Environment

POST http://localhost:3000/api/v3/apel/assignUserToIncident?channel=APP Save Send

Params Authorization Headers (11) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "channel": "ADMIN",
3   "issueId": "636e41b134c714838469e60e",
4   "userId": "6373472c4b18708a98a4abc8",
5   "userId": "assignedTo",
6   "username": "Chazitha"
7 }
8
9
```

Body Cookies Headers (13) Test Results Status: 403 Forbidden Time: 21 ms Size: 543 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Expected completion date should not be past date",
3   "errorCode": 1007
4 }
```

81°F Cloudy Search 18:21 03-05-2023

5. Testing

System tests are designed to validate a fully developed system to assure that it meets its requirements. The test cases are therefore designed solely based on the SRS document.

(OR)

System testing is nothing but how the customer is going to start using your application and checking whether everything is up to the mark and meeting the needs of the customer

5.1 Unit Testing:

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules. Reduces Defects in the Newly developed features or reduces bugs when changing the existing functionality.

Improves design and allows better refactoring of code. Unit Tests, when integrated with build gives the quality of the build as well. It is the first level of functional testing. Below are the test cases on the individual modules of the designed website. The functionality of each module has been checked by the developer of the module.

5.2 Integration Testing:

Integration testing is the second level of the software testing process comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units. Unit testing uses modules for testing purpose, and these modules are combined and tested in integration testing. The goal of integration testing is to check the correctness of communication among all the modules. It includes four types of approaches.

A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated.

Integration testing includes various approaches like

1. Bigbang approach
2. Top down approach
3. Bottom up approach
4. Sandwiched Integration testing.

Conclusion and References

6.1 Conclusion

This project main goal is to serve the people. Roads are one of the main assets of a country where the network infrastructure offers economic and social benefits. It is understood that road maintenance would improve economic development and social benefits, and the poorly maintained road network would destruct mobility, raise accident rates, intensify isolation, poverty and vehicle operating costs. Unidentified road environment covers many aspects and based on that matter, it is important to improve the quality of inspection reports, whereby the efficiencies by mobile application shall be utilized. Developed system mobile apps have been found to be effective, versatile to use, as they can be used anywhere and at any time. The goal was therefore to recognize the types of defects, issues and deficiencies in current road maintenance practices and to develop a computerized highway maintenance program. The findings showed that road maintenance work is not adequately and effectively designed because it requires more time and includes a lot of paperwork. Due to this matter, the mobile application developed by cube highways would provide an invaluable added public and government for easy maintenance.

6.2 References

For adding this features(tasks) in the project, I had referred the following

<https://www.mongodb.com/>

<https://www.w3schools.com/nodejs/>

<https://www.w3schools.com/mongodb/>