PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology (Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of Bachelor of Technology in Information Technology

By

Pallavi Kumari
REG. NO. 39120078

DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI– 600119
www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **PALLAVI KUMARI (Reg. No: 39120078)** who carried out the project entitled "**CRMS-Portal: A Crime Report Management System**" under my supervision from February 2022 to April 2022.

Internal Guide
Dr. Jeberson Retna Raj,

Submitted for Viva voce Examination held on ____________________

Internal Examiner **External Examiner**

DECLARATION

I, **PALLAVI KUMARI** hereby declare that the project report entitled CRMS-Portal: A Web-application for Police authority to report and store E-FIR done by me under the guidance of **Dr. Jeberson Retna Raj** is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Information Technology.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE
PALLAVI KUMARI

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D**, **Dean**, School of Computing, **Dr. R. Subhashini, M.E., Ph.D. Head of the Department** of **Information Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. S. Gowri,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of INFORMATION TECHNOLOGY** who were helpful in many ways for the completion of the project.

ABSTRACT

CRMS-Portal: A Crime record management is an approach towards make First Information Report(FIR) virtual, easy to access and store, paperless. We can see that technology has touched many spheres of our lives in India. There is technology in socializing and maintaining human relations, in and almost every part of our lives. only major part of our society that still remains majorly devoid is the Indian Police Department.

When population was far less, and the crime rates were also comparably minimal. But in today’s India, when population is raising and so many cases being registered every day, it has become a very tedious task to manage the case and all its related documents, manually with the traditional “pen and paper” method. Digitization in Police department is the need of the hour. Hence an E-FIR system will help authority to manage FIR Data virtually.

CRMS-Portal web page will include the reporting system for Police authority where the system at any point of time can provide the details of FIR document, existing charge sheets and their statuses. A database system in which police will keep the record of criminals who have been arrested.

Above platform is created using HTML, CSS, bootstrap, Vanilla JavaScript, NodeJS, Express JS, ReactJS, MongoDB/Mongoose (DATABASE)

CRMS-Portal is another small step in the same direction to help Police Authority to easy access and report of FIR and provide benefits of technology.

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

We can define computing to mean any goal-oriented activity requiring, benefiting from, or creating computers.

Thus, computing includes designing and building hardware and software systems for a wide range of purposes, processing, structuring and managing various kinds of information, doing scientific studies using computers, making computer systems behave intelligently, creating and using communications and entertainment media; finding and gathering information relevant to any particular purpose, and so on. The list is virtually endless, and the possibilities are vast.

So going with the flow and considering the need/our use-case (that is to build a E-FIR reporting system) here we are going to talk about the usage of the service and its build from scratch.

CHAPTER 2: AIM AND SCOPE OF THE PRESENT INVESTIGATION

CRMS-Portal aims at creating a comprehensive and integrated system for enhancing the efficiency and effective policing at all levels and especially at the Police Station level through adoption of technology.

Make the Police functioning citizen friendly and more transparent by automating the functioning of Police Stations. Provide the Investigating Officers of the Civil Police with tools, technology and information to facilitate investigation of crime and detection of criminals.

It can be used to record paper less criminal's record and investigation. This project is mainly useful for adding and getting criminal's record with the all the important documents related to the specific criminal easily and faster with crime ID or with criminal's name.

The authority can preserve records of the criminals and search for any criminal using the system.

This is an online web application with a database system in which police will keep the record of criminals who have been arrested or has got complaint against them.

Improve delivery of citizen-centric services through effective usage of Information and Communication Technology. Keep track of the Progress of Cases, including in Courts.

The following are the advantages of E-police system over the present manual system:

1. **Secure and Transparent Process of Investigation and Tracking:** Since only the investigating officer can access the particular FIR id, the information is private and secure. The process carrying out online, in full knowledge of the complainant ensures transparency.

2. **Improving the standards of Indian Police system:** With many countries like USA, Singapore and many other developed countries in the world already having a fully functional e-police system, India must also develop up to world standards.

3. **No delays in catering the FIR:** As the police has to directly update the complainant over the application about the proceedings of the case, with proof, any delay in the work is instantly noticed by the citizens and thus the scopes of false promises is highly reduced.

4. **Promotion of E-Governance:** With the recent advancement of Creation and Maintenance of police Database, Indian government is now planning to maintain database of 1.5 Crore criminals. The E-Police System will be an additional facility and will aid this process of record maintenance with e-documents.

5. **Ease of Accessibility for authority:** Police authority can access the record any-time and any-where with required authentication details.

6. **Time and Energy Saving:** Using the Web application police authority can fill the form easily and register the complaint efficiently and no need of managing lots of paper and files.

CHAPTER 3: MATERIALS AND METHODS USED

MERN stands for MongoDB, Express, React, Node, after the four key technologies that make up the stack.

- MongoDB - document database
- Express (JS) - Node.js web framework
- React (JS) - a client-side JavaScript framework
- Node (JS) - the premier JavaScript web server

Express and Node make up the middle (application) tier. Express.js is a server-side web framework, and Node.js the popular and powerful JavaScript server platform. Regardless of which variant you choose, ME(RVA)N is the ideal approach to working with JavaScript and JSON, all the way through.

WORKING OF MERN STACK

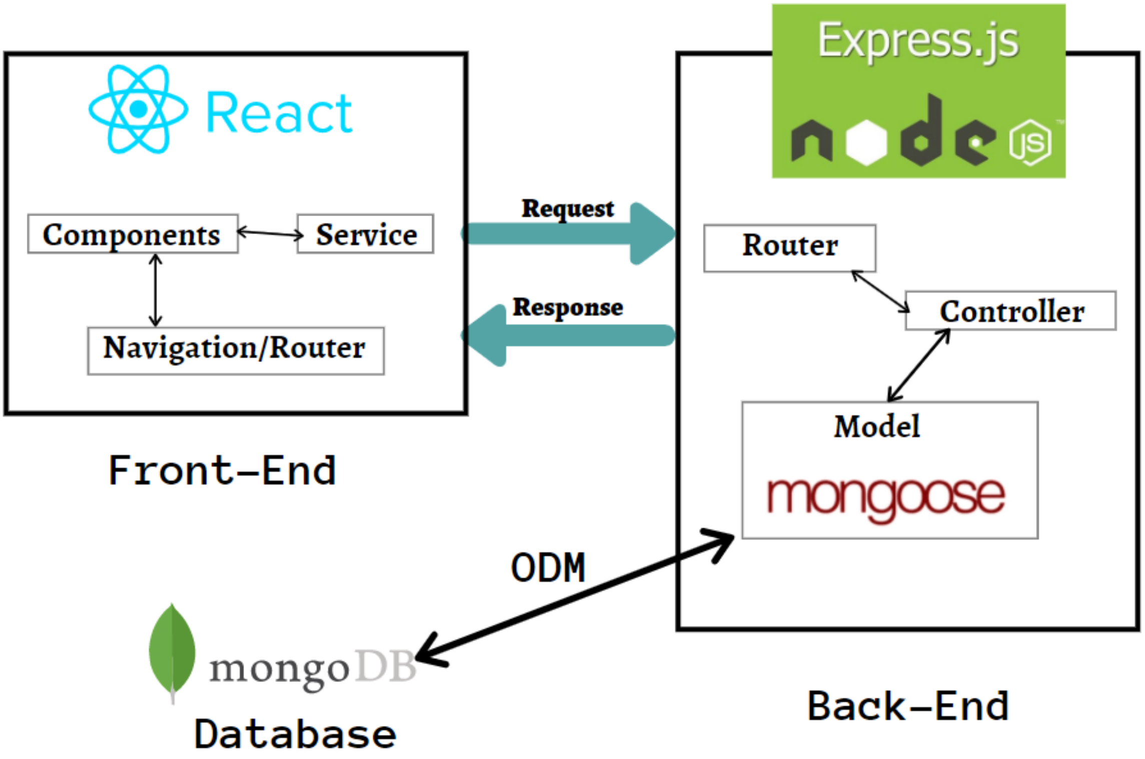


Fig 1.1: MERN stack architecture

React.js Front End

The top tier of the MERN stack is React.js, the declarative JavaScript framework for creating dynamic client-side applications in HTML. React lets you build up complex interfaces through simple Components, connect them to data on your backend server, and render them as HTML.

React's strong suit is handling stateful, data-driven interfaces with minimal code and minimal pain, and it has all the bells and whistles you'd expect from a modern web framework: great support for forms, error handling, events, lists, and more.

Express.js and Node.js Server Tier

The next level down is the Express.js server-side framework, running inside a Node.js server. Express.js bills itself as a "fast, unopinionated, minimalist web framework for Node.js," and that is indeed exactly what it is. Express.js has powerful models for URL routing (matching an incoming URL with a server function), and handling HTTP requests and responses.

By making XML HTTP Requests (XHRs) or GETs or POSTs from your React.js front-end, you can connect to Express.js functions that power your application. Those functions in turn use MongoDB's Node.js drivers, either via callbacks for using Promises, to access and update data in your MongoDB database.

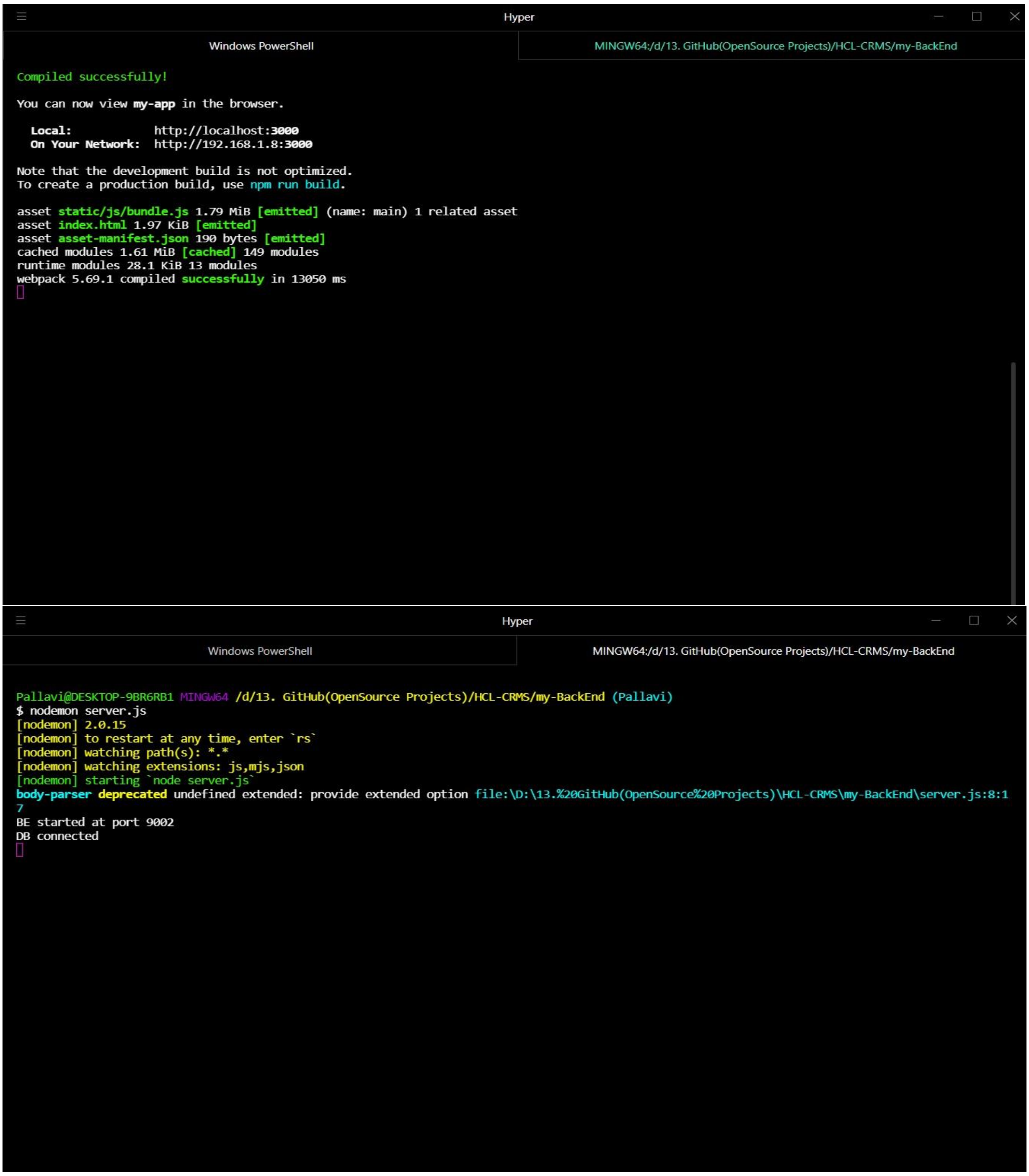
MongoDB Database Tier

If your application stores any data (user profiles, content, comments, uploads, events, etc.), then you're going to want a database that's just as easy to work with as React, Express, and Node.

That's where MongoDB comes in: JSON documents created in your React.js front end can be sent to the Express.js server, where they can be processed and (assuming they're valid) stored directly in MongoDB for later retrieval. Again, if you're building in the cloud, you'll want to look at Atlas. If you're looking to set up your own MERN stack, read on!

CHAPTER 4: WALKTHROUGH OF PRODUCT

STEP 1: Starting the local servers for client-side, server-side and databases and checking for any DDoS Attack before accessing


Hyper
Windows PowerShell
MINGW64:/d/13. GitHub(OpenSource Projects)/HCL-CRMS/my-BackEnd

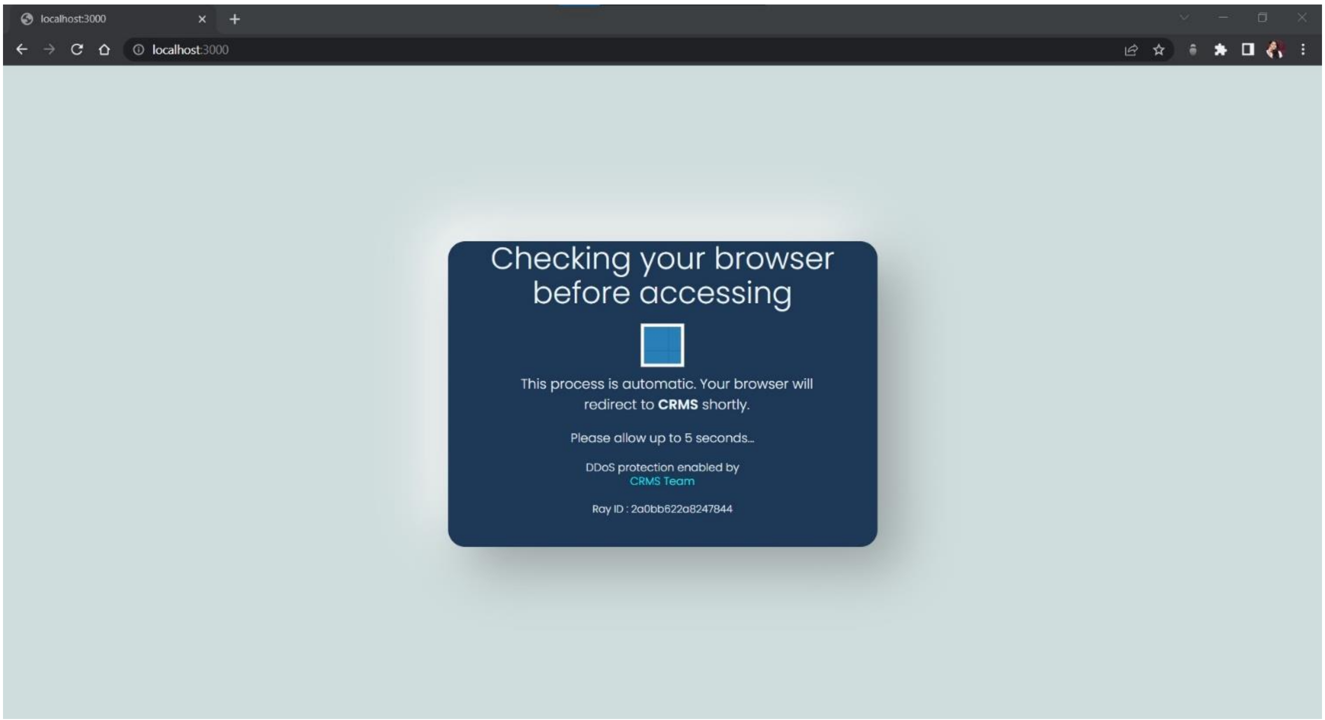
```
Compiled successfully!

You can now view my-app in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.1.8:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

asset static/js/bundle.js 1.79 MiB [emitted] (name: main) 1 related asset
asset index.html 1.97 KiB [emitted]
asset asset-manifest.json 190 bytes [emitted]
cached modules 1.61 MiB [cached] 149 modules
runtime modules 28.1 KiB 13 modules
webpack 5.69.1 compiled successfully in 13050 ms
```

Hyper
Windows PowerShell
MINGW64:/d/13. GitHub(OpenSource Projects)/HCL-CRMS/my-BackEnd

```
Pallavi@DESKTOP-9BR6RB1 MINGW64 /d/13. GitHub(OpenSource Projects)/HCL-CRMS/my-BackEnd (Pallavi)
$ nodemon server.js
[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node server.js`
body-parser deprecated undefined extended: provide extended option file:\D:\13.%20GitHub(OpenSource%20Projects)\HCL-CRMS\my-BackEnd\server.js:8:1
7
BE started at port 9002
DB connected
```

Fig 1.2: Starting local severs

Step-2: Run Localhost at port 3000



Checking your browser before accessing

This process is automatic. Your browser will redirect to **CRMS** shortly.

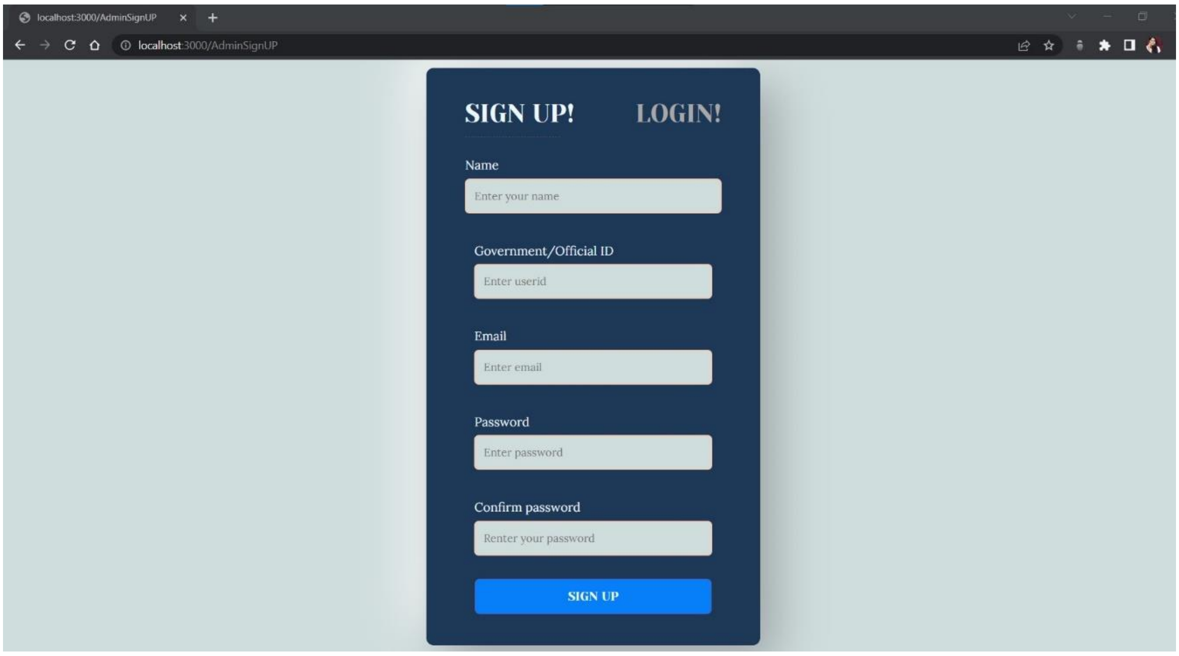
Please allow up to 5 seconds...

DDoS protection enabled by
CRMS Team

Ray ID : 2a0bb622a8247844

Fig 1.3: DDoS attack check

Step – 3: The Admin Sign Up and Admin Login Page to Authenticate User:



Name
Enter your name

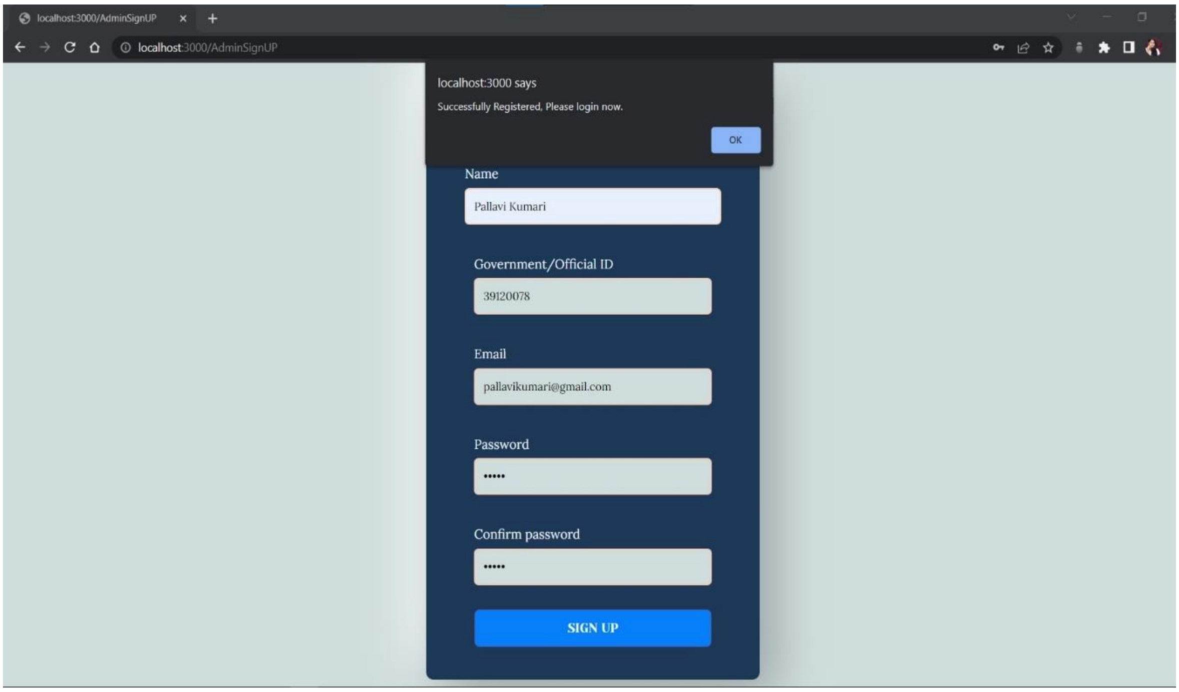
Government/Official ID
Enter userid

Email
Enter email

Password
Enter password

Confirm password
Renter your password

Fig 1.4: Sign-up page



localhost:3000 says

Successfully Registered, Please login now.

Name: Pallavi Kumari

Government/Official ID: 39120078

Email: pallavikumari@gmail.com

Password: •••••

Confirm password: •••••

Fig 1.5: Sign Up Confirmation

> After successful account creation one can carry on with the login and later submit the FIR Report

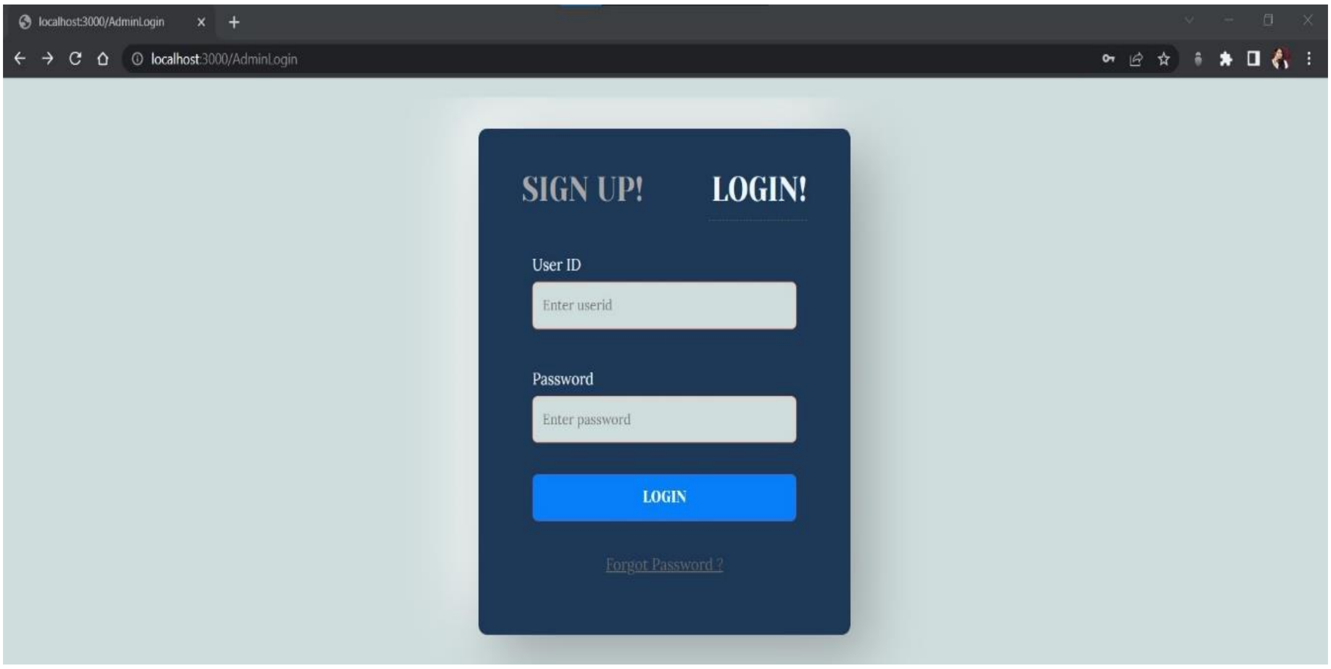
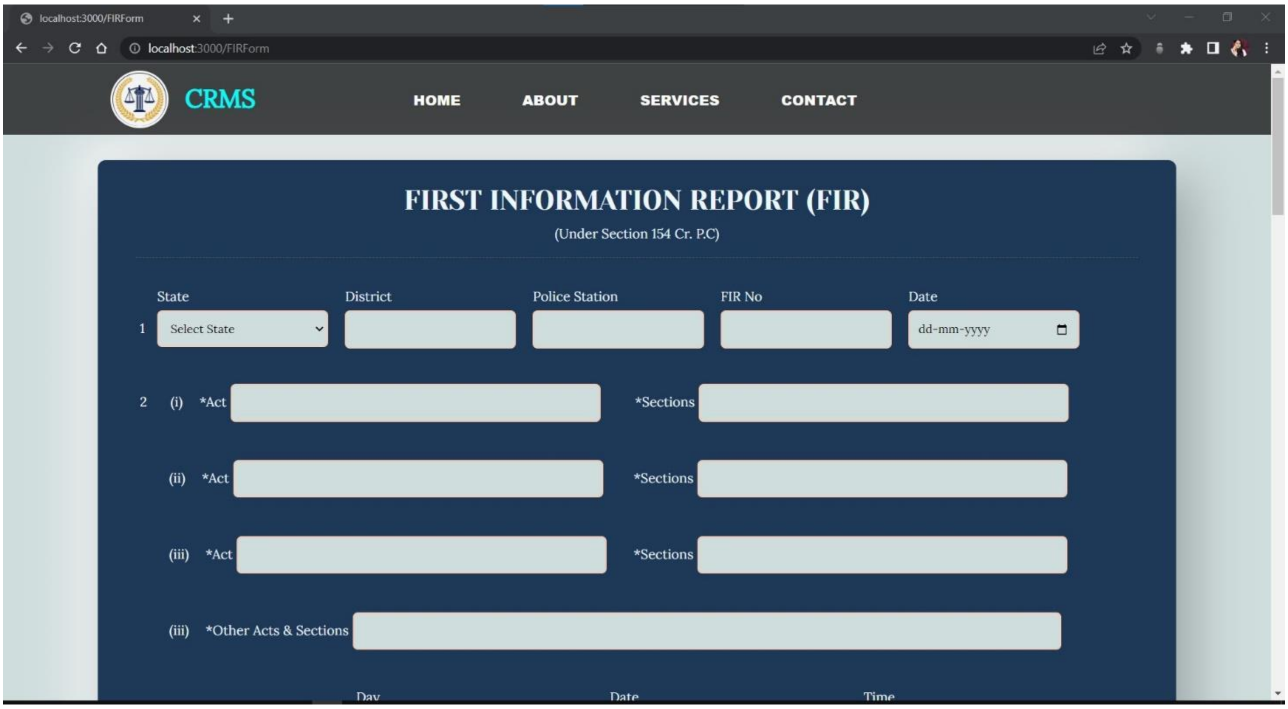


Fig 1.6: Log In Page

>After successful Login move to the Home page/ FIR Form



FIRST INFORMATION REPORT (FIR)

(Under Section 154 Cr. P.C)

1 State: Select State | District: | Police Station: | FIR No: | Date: dd-mm-yyyy

2 (i) *Act | *Sections

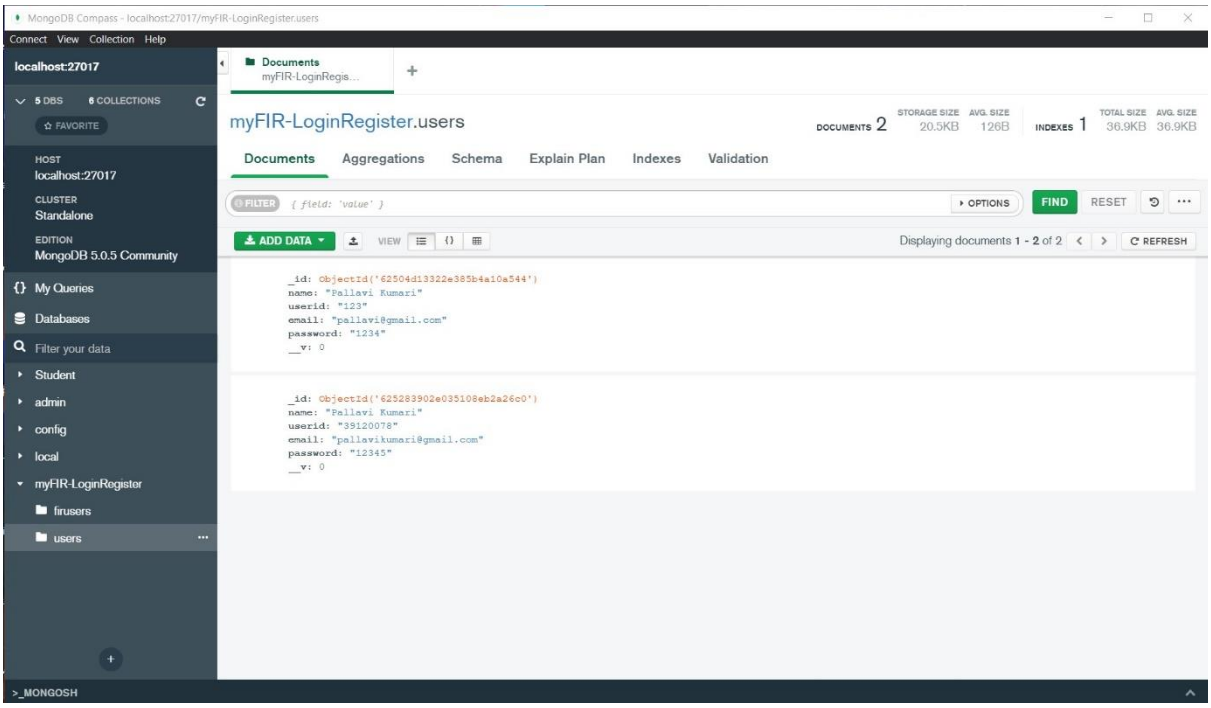
(ii) *Act | *Sections

(iii) *Act | *Sections

(iii) *Other Acts & Sections

Day | Date | Time

Fig 1.7: Home page/the bug ticketing page



HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 5.0.5 Community

myFIR-LoginRegister.users

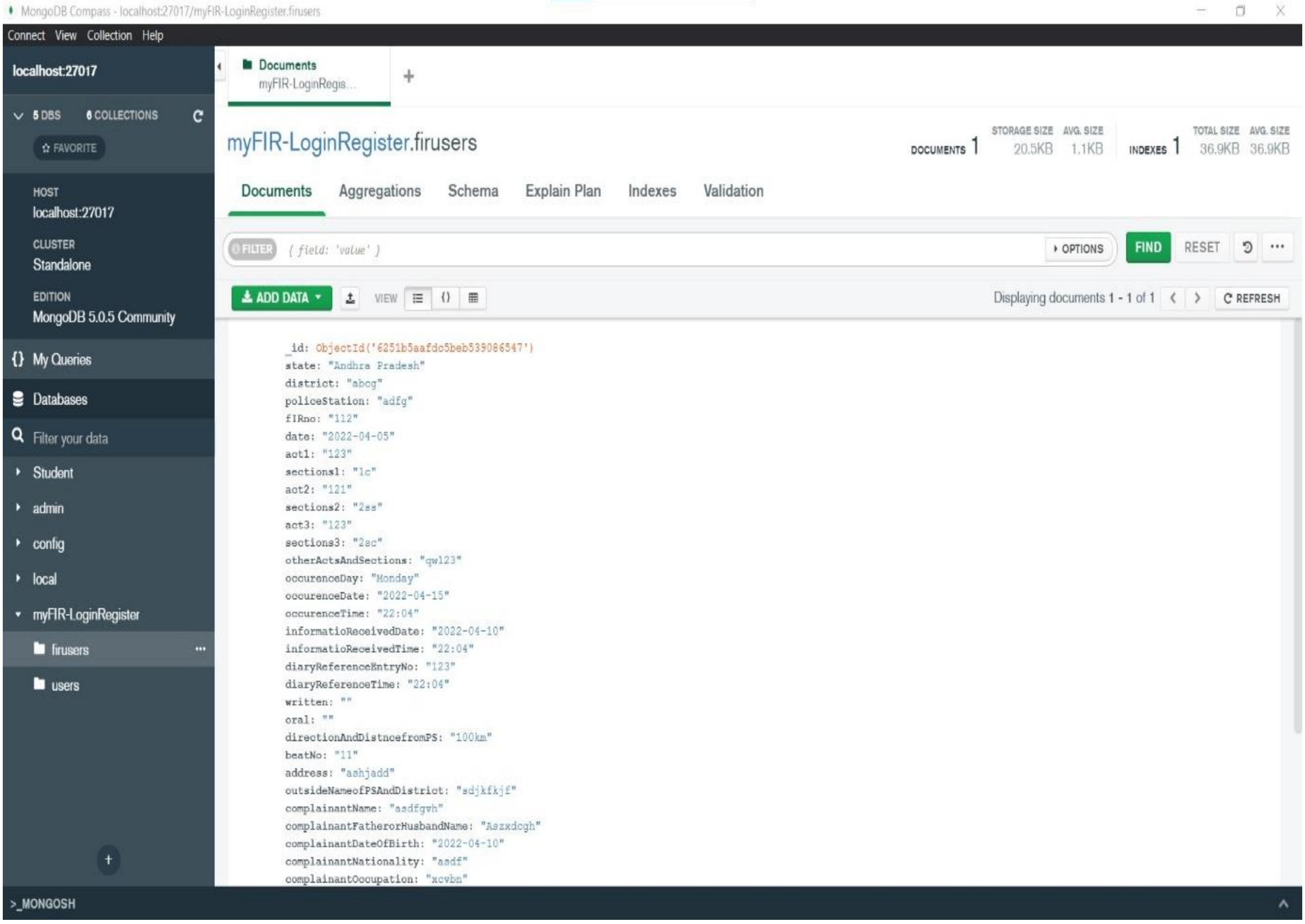
DOCUMENTS 2 STORAGE SIZE 20.5KB AVG. SIZE 126B INDEXES 1 TOTAL SIZE 36.9KB AVG. SIZE 36.9KB

{ field: 'value' }

Displaying documents 1 - 2 of 2

```
_id: ObjectId('62504d13322e385b4a10a544')
name: "Pallavi Kumari"
userid: "123"
email: "pallavi@gmail.com"
password: "1234"
__v: 0

_id: ObjectId('625283902e035108eb2a26c0')
name: "Pallavi Kumari"
userid: "39120078"
email: "pallavikumari@gmail.com"
password: "12345"
__v: 0
```

Fig 1.8: Log-Register Database console



```
_id: ObjectId('6251b5aafdc5beb539086547')
state: "Andhra Pradesh"
district: "abcg"
policeStation: "adfg"
fIRno: "112"
date: "2022-04-05"
act1: "123"
sections1: "1c"
act2: "121"
sections2: "2ss"
act3: "123"
sections3: "2sc"
otherActsAndSections: "qw123"
occurenceDay: "Monday"
occurenceDate: "2022-04-15"
occurenceTime: "22:04"
informatioReceivedDate: "2022-04-10"
informatioReceivedTime: "22:04"
diaryReferenceEntryNo: "123"
diaryReferenceTime: "22:04"
written: ""
oral: ""
directionAndDistncefromPS: "100km"
beatNo: "11"
address: "ashjadd"
outsideNameofPSAndDistrict: "sdjkfkjf"
complainantName: "asdfgvh"
complainantFatherorHusbandName: "Aszxdcgh"
complainantDateOfBirth: "2022-04-10"
complainantNationality: "asdf"
complainantOccupation: "xcvbn"
```

Fig 1.9: FIR Form Database console

CHAPTER 5: RESULT AND DISCUSSION

After following through all the above process and instruction we finally will be able to access the live full stack website through the localhost link generated.

SERVER-SIDE CODE IS GIVEN BELOW:

```
import express from "express"
import cors from "cors"
import mongoose from "mongoose"
// const FIRData = require('/FIRformServer');

const app = express()
app.use(express.json())
app.use(express.urlencoded())
app.use(cors())

mongoose.connect("mongodb://localhost:27017/myFIR-LoginRegister", {
    useNewUrlParser: true,
    useUnifiedTopology: true
}, () => {
    console.log("DB connected")
})



//////////////////////////   REGISTER-LOGIN DATABASE SCHEMA ////////////////////////////////



const userSchema = new mongoose.Schema({
    name: String,
    userid: String,
    email:String,
    password: String
})

const User = new mongoose.model("User", userSchema)

// Post Routes
app.post("/AdminLogin", (req, res)=> {
  console.log(req.body)
```

```
    const {userid, password} = req.body
    User.findOne({ userid: userid}, (err, user) => {

        if(user){
            if(password === user.password ) {
                res.send({message: "Login Successfull", user: user})
                console.log(user);
            } else {
                res.send({ message: "Password didn't match"})
            }
        } else {
            res.send({message: "User not registered"})
        }
    })
})


app.post("/AdminSignUP", (req, res)=> {
  console.log(req.body);
    const { name, userid, email, password} = req.body
    User.findOne({userid: userid}, (err, user) => {
        if(user){
            res.send({message: "User already registerd"})
        } else {
            const user = new User({
                name,
                userid,
                email,
                password
            })
            user.save(err => {
                if(err) {
                    res.send(err)
                } else {
                    res.send( { message: "Successfully Registered, Please login now." })
                }
            })
        }
    })

})

//////////////////////////////////////////////////////    END     //////////////////////////////////////////////////


///////////////////////////////////////////   FIR FORM DATABASE SCHEMA ///////////////////////

const fIRSchema = new mongoose.Schema({
  state: String,
  district: String,
  policeStation: String,
```

```
  fIRno: String,
  date: String,
  act1: String,
  sections1: String,
  act2: String,
  sections2: String,
  act3: String,
  sections3: String,
  otherActsAndSections: String,

  occurenceDay: String,
  occurenceDate: String,
  occurenceTime: String,
  informatioReceivedDate: String,
  informatioReceivedTime: String,
  diaryReferenceEntryNo: String,
  diaryReferenceTime: String,

  written: String,
  oral: String,

  directionAndDistncefromPS: String,
  beatNo: String,
  address: String,
  outsideNameofPSAndDistrict: String,

  complainantName: String,
  complainantFatherorHusbandName: String,
  complainantDateOfBirth: String,
  complainantNationality: String,
  complainantOccupation: String,
  complainantPassportNo: String,
  complainantDateofIssue: String,
  complainantPlaceOfIssue: String,
  complainantAddress: String,

  detailsOfSuspected: String,
  reasonsforDelay: String,
  particularsOfPropertiesStolenInvolved: String
})



const FIRUser = new mongoose.model("FIRUser", fIRSchema)

app.post("/FIRform", (req, res)=> {
  console.log(req.body);
    const { state, district, policeStation, fIRno, date, act1, sections1, act2, sections2, act3,
sections3, otherActsAndSections, occurenceDay, occurenceDate, occurenceTime,
informatioReceivedDate, informatioReceivedTime,
```

```
        diaryReferenceEntryNo, diaryReferenceTime, written, oral,
directionAndDistncefromPS, beatNo, address, outsideNameofPSAndDistrict,
complainantName, complainantFatherorHusbandName, complainantDateOfBirth,
complainantNationality,
        complainantOccupation, complainantPassportNo, complainantDateofIssue,
complainantPlaceOfIssue, complainantAddress, detailsOfSuspected, reasonsforDelay,
particularsOfPropertiesStolenInvolved} = req.body
  FIRUser.findOne({fIRno: fIRno}, (err, firDetails) => {
    if(firDetails){
      res.send({message: "FIR Number already registerd"})
    } else {
      const firDetails = new FIRUser({
       state,
       district,
       policeStation,
       fIRno,
       date,

       act1,
       sections1,
       act2,
       sections2,
       act3,
       sections3,
       otherActsAndSections,

       occurenceDay,
       occurenceDate,
       occurenceTime,
       informatioReceivedDate,
       informatioReceivedTime,
       diaryReferenceEntryNo,
       diaryReferenceTime,

       written,
       oral,

       directionAndDistncefromPS,
       beatNo,
       address,
       outsideNameofPSAndDistrict,

       complainantName,
       complainantFatherorHusbandName,
       complainantDateOfBirth,
       complainantNationality,
       complainantOccupation,
       complainantPassportNo,
       complainantDateofIssue,
       complainantPlaceOfIssue,
       complainantAddress,
```

```
       detailsOfSuspected,
       reasonsforDelay,
           particularsOfPropertiesStolenInvolved
         })
         firDetails.save(err => {
            if(err) {
               res.send(err)
            } else {
               res.send( { message: "fIR Record Added Successfully" })
            }
         })
      }
   })

})

//////////////////////////////////////////////////////////  END   //////////////////////////////////////////////////////////////////


app.listen(9002,() => {
   console.log("BE started at port 9002")
})
```

CHAPTER 6: SUMMARY AND CONCLUSION

There are numerous architectural and conceptual considerations when you are contemplating migrating our web application/ website to the localhost. Indian Police System has remained devoid of web technology, with most works being carried out on a pen and paper basis. This traditional method is prone to delays and inefficiency. This paper proposes to simplify and speed up the process of FIR registration and tracking. With the advancement and incorporation of internet and web technology into the Indian Police System, it will definitely boost up the proceedings. This project aims to help the police officers. The updates about case details are notified directly will be accessible online. The ease of access of the web application by the police of India will encourage a more judicial and lawful society.