

CERTIFICATE

This is to certify that the project entitled

“Complaint Box”

Has been satisfactorily completed by

- 1) Nadim Ayubkhan Sanadi
- 2) Vaishnavi Prabhakar Patil
- 3) Nishant Babarao Dhabe
- 4) Rushikesh Narayan Patil
- 5) Omkar Somnath Kokane

Under super vision and guidance for partial fulfillment of the
PG Diploma in Advanced Computing (DAC) (24 weeks Full Time) Course
of
Centre for Development of Advanced Computing (C-DAC), Pune.
at
Academy of Information Technology (YCP)
Nariman Point, Mumbai –400 021.

Faculty

Course Co-ordinator

Batch: PG DAC March 2024

Date: August 2024.

PREFACE

In today's interconnected society, the need for effective tools to address public safety and enhance communication between citizens and law enforcement is more pressing than ever. The Complaint Box project is designed to bridge this gap by offering a streamlined, accessible platform that transforms the way complaints are lodged, managed, and resolved.

The Complaint Box serves as a centralized hub where users can effortlessly report criminal activities, locate nearby police stations, and track the status of their complaints. This platform is more than just a complaint management system; it is a vital tool that empowers citizens to contribute to public safety, ensuring that every concern is addressed promptly and effectively.

Gone are the days of navigating through complex processes to file a complaint. With the Complaint Box, users can quickly and easily submit their reports, knowing that their voices will be heard and their issues will be addressed. The system's intuitive interface and robust backend make the complaint process as simple as a few clicks, allowing for faster response times and more efficient case investigations.

But the Complaint Box is not just a platform for reporting incidents—it's a community-driven initiative that fosters a safer, more connected society. By facilitating communication and information sharing between the public and law enforcement, the platform helps build trust and accountability, ensuring that every citizen's concerns are taken seriously.

Join us in this endeavor as we transform the way complaints are handled, making our communities safer and more responsive. The Complaint Box project is where technology meets public service, creating a structured, transparent, and effective system for managing and resolving complaints.

ACKNOWLEDGMENT

We extend our deepest gratitude to the many individuals and teams whose invaluable contributions have made the Complaint Box Project a reality. Our dedicated team of developers, UX/UI designers, and project managers has tirelessly committed their expertise, creativity, and passion to building a platform that truly addresses the needs of users seeking to voice and resolve their concerns. Their steadfast dedication and attention to detail have been crucial in transforming our vision into a functional and user-friendly tool.

We are profoundly thankful to our users, whose insightful feedback and active participation have been at the heart of this project. Your input has been instrumental in guiding our development process, helping us to refine features and ensure that the Complaint Box meets the highest standards of usability and effectiveness. Your engagement has inspired us to continuously strive for excellence and to remain responsive to the needs of the community.

Moreover, we would like to extend our sincere appreciation to our partners and collaborators. Your expertise, resources, and unwavering support have been pivotal in bringing this project to fruition. The collaborative efforts between our teams have not only enhanced the technical and operational aspects of the Complaint Box but have also fostered a shared commitment to improving the way complaints are managed and resolved.

As we look toward the future, we are excited about the potential for the Complaint Box to grow and evolve. We remain committed to our mission of creating a more transparent, accountable, and user-centric environment for addressing grievances. Thank you to everyone who has walked this journey with us, providing the encouragement and assistance needed to make this project successful. Your involvement has been invaluable, and we look forward to continuing this journey together.

Lastly, we would like to express our heartfelt thanks to our families and friends. Your patience, encouragement, and understanding throughout the development of this project have been a constant source of motivation. Your belief in our vision has sustained us through challenges, and your support has been a cornerstone of our success.

Complaint Box

To everyone who has played a role ,big or small, in making this project a reality – thank you from the bottom of our hearts.

Team Members;

1. Nadim Ayubkhan Sanadi
2. Vaishnavi Prabhakar Patil
3. Nishant Babarao Dhabe
4. Rushikesh Narayan Patil
5. Omkar Somnath Kokane

INDEX

SR.NO	TITLE	PAGENO
1.	Certificate	1
2.	Preface	2
3.	Acknowledgement	3
4.	Introduction	6
5.	Software Requirement	7
6.	Hardware Requirement	7
7.	ER Diagram	8
8.	Data Flow Diagram	9
9.	Use Case Diagram	10
10.	Backend Details	11
11.	Frontend Details	12
	Interface: <ul style="list-style-type: none"> • Landing Page: • Super Admin login Page: • Admin Login Page: • Admin Details Page: • Complaint Details Page: • Lodge Complaint Page: • Card Page: • From Page: 	13-16
12.	Front End Coding	17-21
13.	Back End Coding	22-25
14.	Future Scope	26
15.	Conclusion	27
16.	Bibliography	28
17.		

INTRODUCTION

In an era defined by technological advancements and rapid digital transformation, public safety stands as a cornerstone of community well-being. The Complaint Box project emerges as a beacon of change, revolutionizing traditional methods of reporting criminal activity and connecting citizens with law enforcement. Imagine a platform that not only facilitates communication and information sharing but also enhances public safety through a streamlined and efficient reporting system – that's what the Complaint Box is all about.

The Complaint Box is more than just a Web Application; it's a bridge between citizens and law enforcement. It offers a centralized platform designed to manage and receive reports, ensuring that critical information is processed swiftly and accurately. From locating police stations to filing and managing complaints, the Complaint Box streamlines the process, allowing users to focus on what truly matters – ensuring their safety and that of their community.

One of the standout features of the Complaint Box is its user-friendly interface, which adds a unique dimension to the platform. Users can easily report criminal activities, share essential information, and track the progress of their complaints, creating a connected and informed community. The Complaint Box transcends traditional reporting methods, turning them into a tech-savvy, structured platform where public safety is not just a priority but also a shared responsibility.

Join us on this critical journey as we transform public safety into a dynamic and interactive experience. Together, we are redefining the future of community protection, one report at a time. Welcome to the Complaint Box project, where we are shaping the future of public safety, one interaction at a time.

Complaint Box

Software Requirements:

- Visual Studio
- Visual Studio Code
- My SQL
- Browser

Hardware Requirements:

- PC/Laptop
- I5 Processor minimum
- 8GB Ram

Tools Requirements:

- Swagger
- Git
- GitHub

ER Diagram

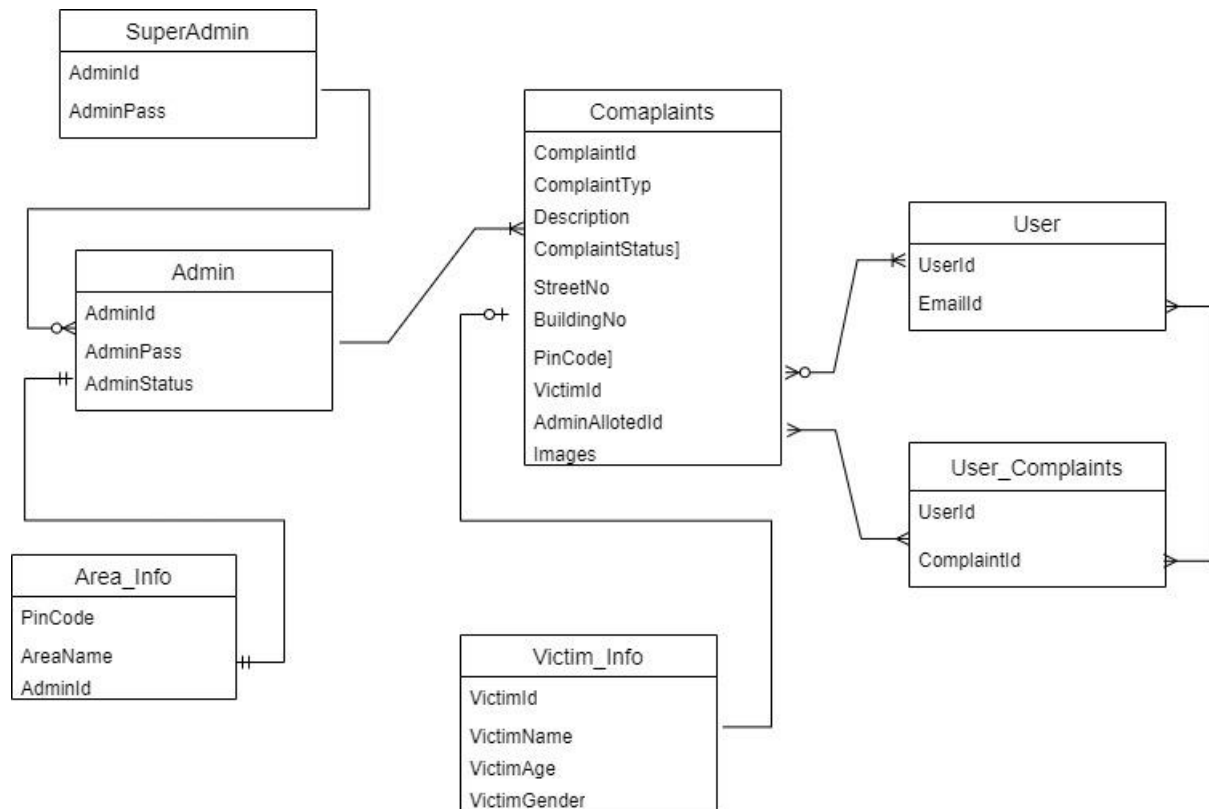


Diagram-1:ER-Diagram

Data Flow Diagram

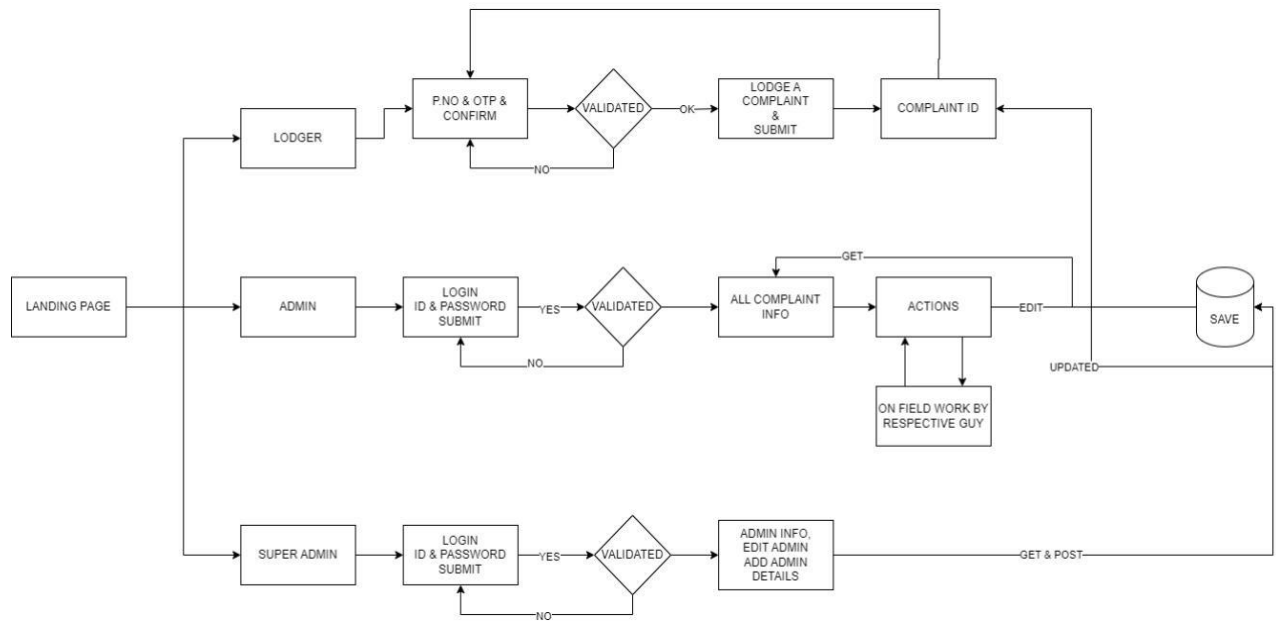


Diagram-2: Data Flow Diagram

Use case Diagram

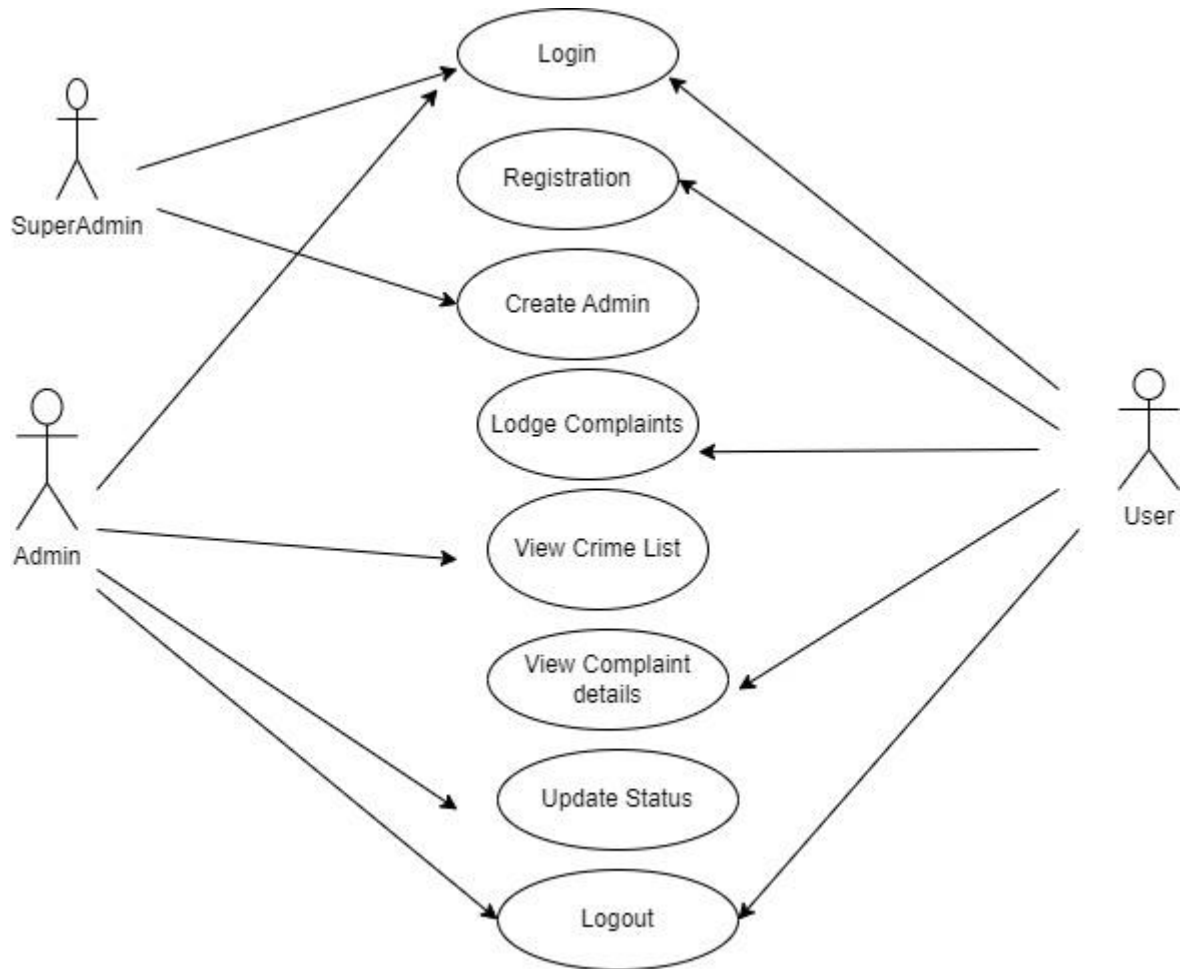


Diagram-3:Use Case Diagram

BACKENDDetails

What technology used?

- Asp Dot Net
- My SQL (Database)
- Micro service

Dependencies:

API End Points:

- User Login Functionality
- Admin Login Functionality

FRONT END DETAILS

React.js, commonly referred to as React, is an open-source Java Script library for building user interfaces (UIs) or user interface components. It was developed and is maintained by Facebook and a community of individual developers and companies. React is often used for building single-page applications and mobile applications.

What libraries are used in React?

React-router-Dom

AXIOS

Super Admin:

Super-Admin can create and assign admin with respect to Area.

Super-Admin can Monitor Activity of Admin.

Admin Login:

Admin can only monitor Complaints

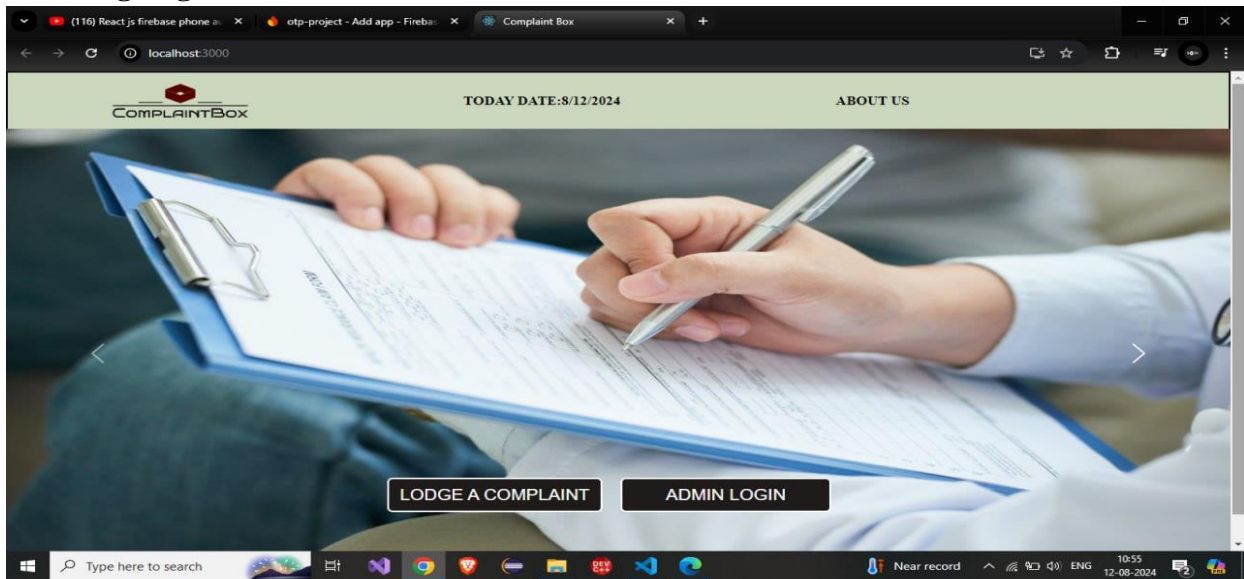
Admin can manage complaint with respect to area

User Login:

This is user Login in page if user can enter their Details in a text box and click the “send OTP” button to proceed with lodging a complaint.

Interface:

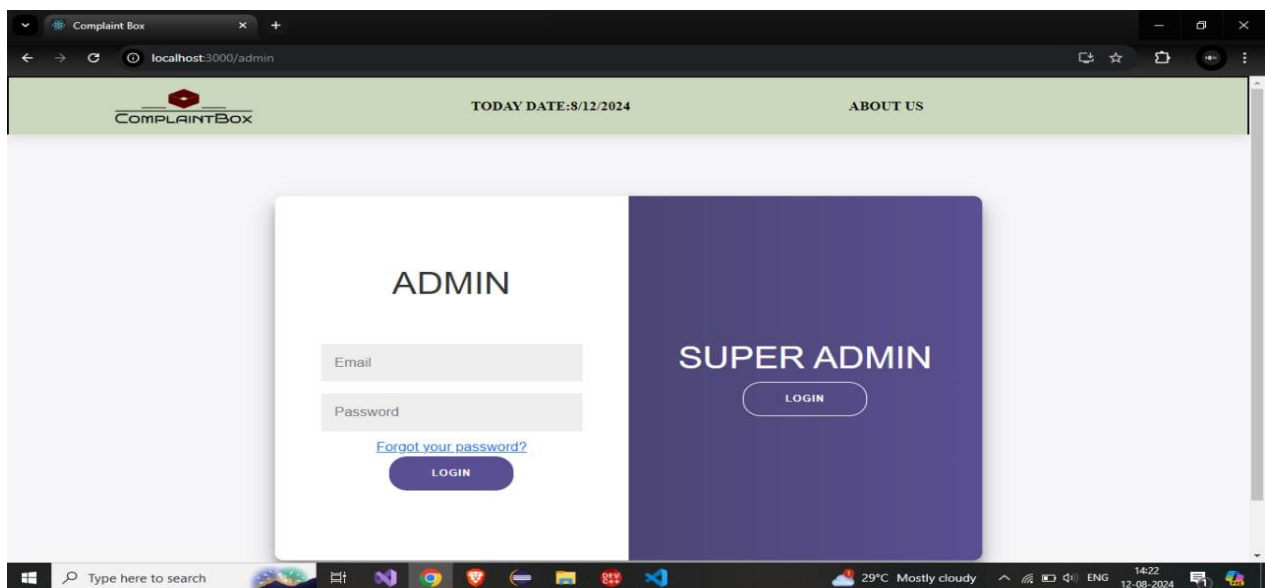
Landing Page:



Screenshot-1

The Landing page of the Complaint Box features a clean and professional interface, allowing users to easily lodge a complaint or access the admin login. The prominent image of someone filling out a form reinforces the platform's focus on facilitating efficient complaint management.

Admin Login Page:



Screenshot-3

The admin login page provides a simple and secure interface for administrators to access the Complaint Box's backend. It offers separate login sections for standard admins and super admins, ensuring that different levels of access are managed effectively.

Complaint Box

Admin Details Page:

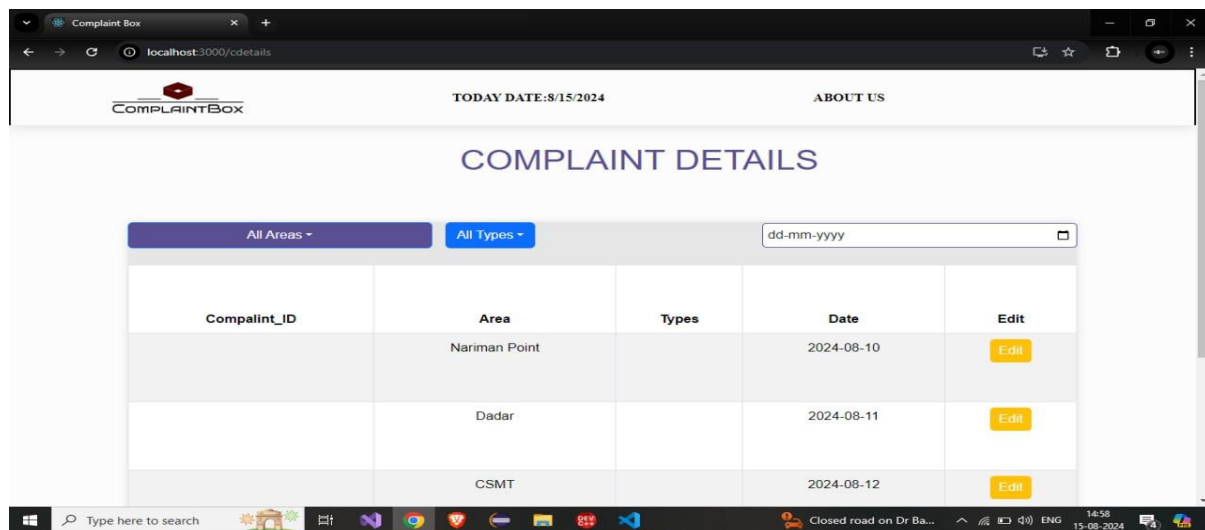
Admin Name	Area	Status	Date	Edit
Admin 1	Nariman Point	Active	2024-08-10	Edit
Admin 2	Dadar	Inactive	2024-08-11	Edit
Admin 3	CSMT	Active	2024-08-12	Edit

Screenshot-4

The "Admin Details" page of the "Complaint Box" application displays a table listing administrators by name, area, status, and date of entry, with options to edit each record. Filters for area and status, as well as an "Add Admin" button, enhance the user management functionality.

Complaint Box

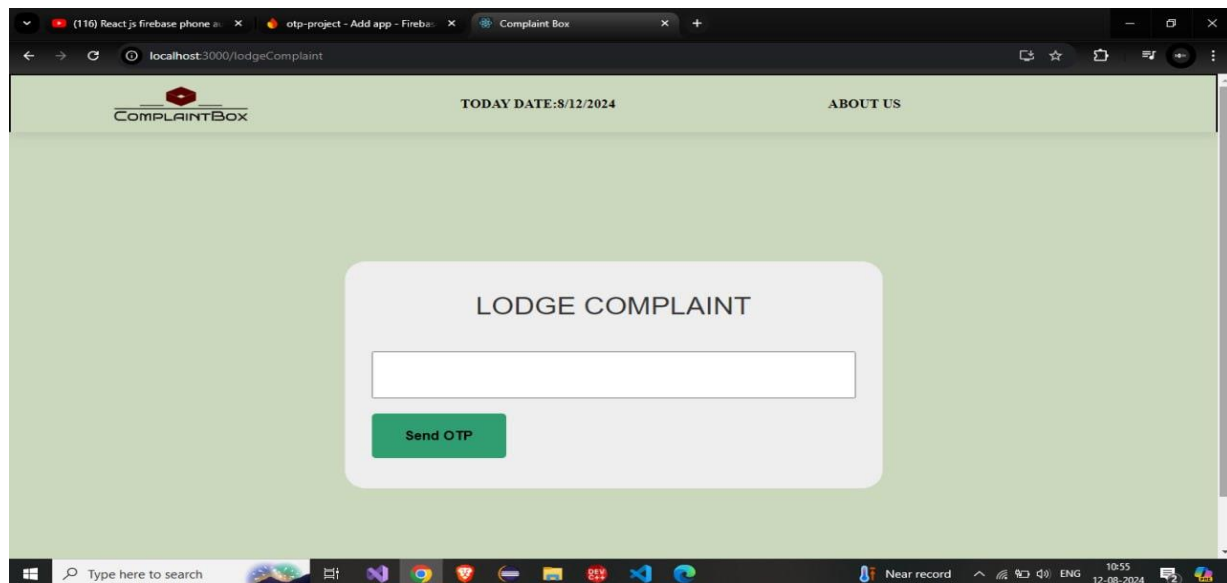
Complaint Details Page:



Screenshot-5

This image shows the "Complaint Details" page of the 'Complaint Box' project, where users can view, filter by area and type, and edit complaints. The page includes a table listing complaints with details such as complaint ID, area, type, date, and an option to edit each complaint.

Lodge Complaint Page:

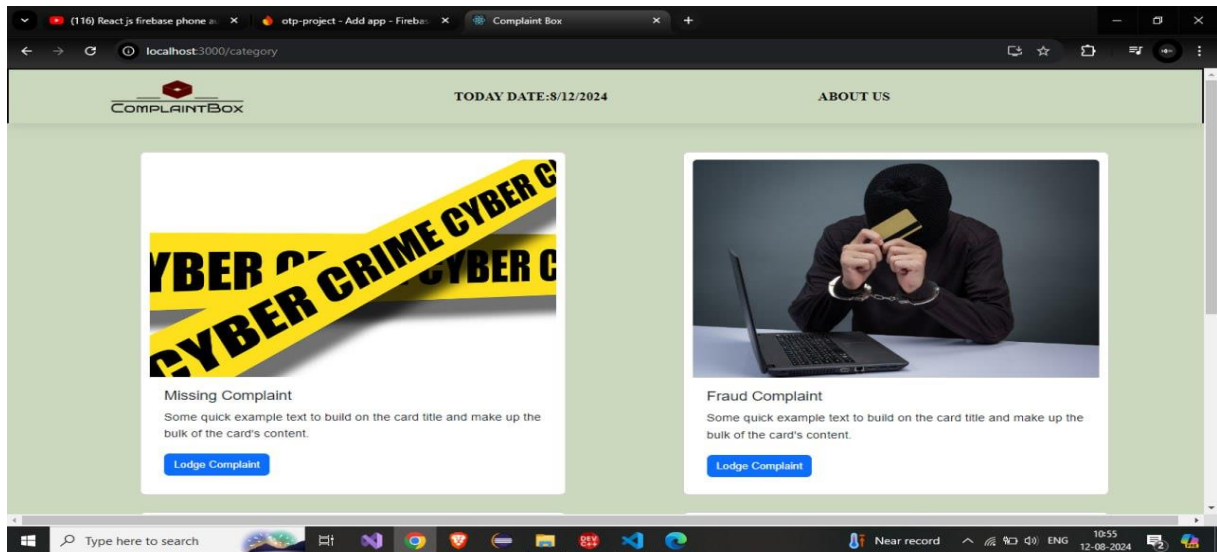


Screenshot-6

This is user Login in page if user can enter their Details in a text box and click the “send OTP” button to proceed with lodging a complaint.

Complaint Box

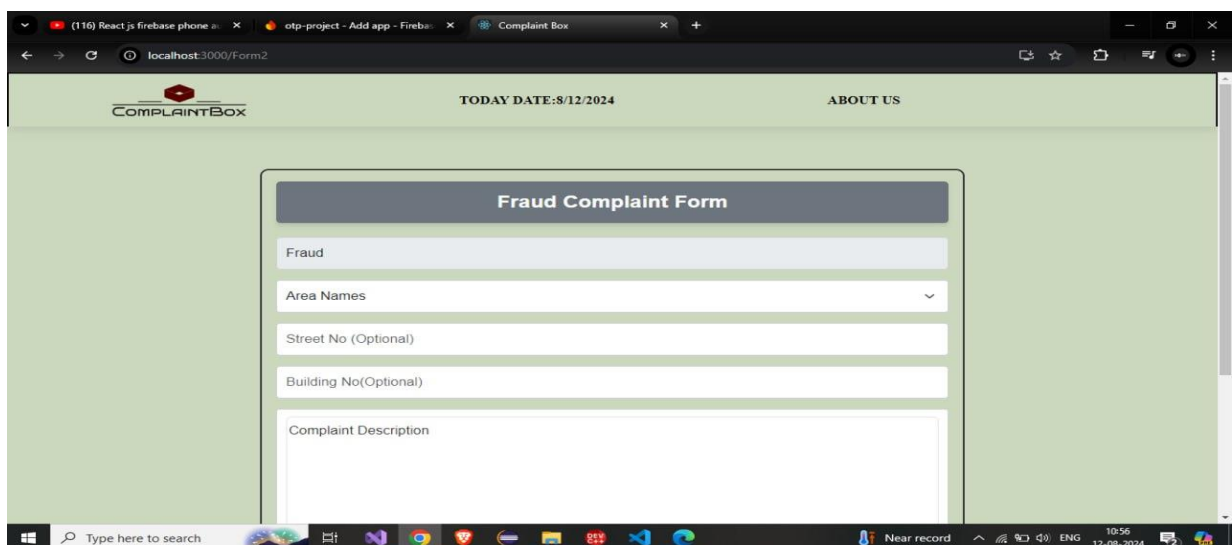
Card page:



Screenshot-7

This image shows the 'Complaint Box' category page with options to lodge different types of complaints, including missing and fraud complaints.

Form pages:



Screenshot-9

The form displayed is titled "Fraud Complaint Form" with fields for area names, street number, building number, and a complaint description.

Complaint Box

From Page:

The screenshot displays a web application running on a local host. The browser window has three tabs: '(116) React.js firebase phone a...', 'otp-project - Add app - Fireba...', and 'Complaint Box'. The address bar shows 'localhost:3000/Form1'. The website header includes the 'COMPLAINTBOX' logo, the text 'TODAY DATE:8/12/2024', and a link 'ABOUT US'. The main content area features a form titled 'Missing Complaint Form'. The form contains a 'Missing' dropdown menu, an 'Area Names' dropdown menu, a 'Street No (Optional)' text input, a 'Building No(Optional)' text input, and a 'Complaint Description' text area. The Windows taskbar at the bottom shows the search bar and several application icons. The system clock in the bottom right corner indicates '10:55' on '12-08-2024'.

Screenshot-10

The form displayed is titled "Missing Complaint Form" with similar fields for area names, street number, building number, and a complaint description.

Front End Coding

Super Admin/Admin :

```
import React, { useRef, useEffect } from "react";
import { Link } from "react-router-dom";

const LoginSignUp = () => {
  const containerRef = useRef(null);

  useEffect(() => {
    const container = containerRef.current;
    const signUpButton = container.querySelector('#signUp');
    const signInButton = container.querySelector('#signIn');

    signUpButton.addEventListener('click', () => {
      container.classList.add("right-panel-active");
    });
  });
}
```

Complaint Box

```
});

signInButton.addEventListener('click', () => {
  container.classList.remove("right-panel-active");
});

// Cleanup event listeners on component unmount
return () => {
  signUpButton.removeEventListener('click', () => {
    container.classList.add("right-panel-active");
  });

  signInButton.removeEventListener('click', () => {
    container.classList.remove("right-panel-active");
  });
};
}, []);

return (
  <div className="login-body1">
    <div className="login-container1" id="login-container" ref={containerRef}>
      <div className="form-container1 sign-up-container1">
        <form action="#" className="frmadm">
          <h1>SUPER ADMIN </h1>
          {/* <input type="text" placeholder="Name" /> */}
          <input type="email" value="dhabenishant@gmail.com" placeholder="Email"
className="inpadm" />
          <input type="password" value="000000" placeholder="Password" className="inpadm" />
          <a href="#">Forgot your password?</a>
          {/* <button className="ghost1">LOGIN</button> */}
          <Link to="/adetails" className="ghost1">LOGIN</Link>

        </form>
      </div>
      <div className="form-container1 sign-in-container1">
```

Complaint Box

```
<form action="#" className="frmadm">
  <h1>ADMIN</h1>
  <input type="email" value="nadim@gmail.com" placeholder="Email"
className="inpadm" />
  <input type="password" value="00000" placeholder="Password" className="inpadm" />
  <a href="#">Forgot your password?</a>
  { /* <button className="ghost1">LOGIN</button> */ }
  <Link to="/cdetails" className="ghost1">LOGIN</Link>
</form>
</div>
<div className="overlay-container1">
  <div className="overlay1">
    <div className="overlay-panel1 overlay-left">
      <h1>ADMIN</h1>

      <button className="ghost1" id="signIn">LOGIN</button>
    </div>
    <div className="overlay-panel1 overlay-right">
      <h1>SUPER ADMIN </h1>

      <button className="ghost1" id="signUp">LOGIN</button>
    </div>
  </div>
</div>
</div>
</div>
);
};
```

```
export default LoginSignUp;
```

Landing page:

```
import Carousel from 'react-bootstrap/Carousel';
import { Link } from 'react-router-dom';
```

Complaint Box

```
var heroData = [
  {
    id: 1,
    image: require('../assets/images/img7.jpg'),
  },
  {
    id: 2,
    image: require('../assets/images/img6.jpg'),
  },
  {
    id: 3,
    image: require('../assets/images/img5.jpg'),
  }
]

export default function CarouselComp() {
  return (
    <section id="home" className='hero-block'>
      <Carousel>
        {
          heroData.map(hero => {
            return (
              <Carousel.Item key={hero.id}>
                <img
                  className="d-block w-100"
                  src={hero.image}
                  alt={"Slide" + hero.id}
                />
                <Carousel.Caption>
                  <Link to="/lodgeComplaint" className='btn btn-primary custom-btn mb-5 fs-5'>Lodge A Complaint</Link>
                  <Link to="/admin" className='btn btn-primary custom-btn mb-5 fs-5'>Admin Login</Link>
                </Carousel.Caption>
              </Carousel.Item>
            )
          })
        }
      </Carousel>
    </section>
  )
}
```

Complaint Box

```
        </Carousel.Caption>
      </Carousel.Item>
    )
  })
}
</Carousel>
</section>
)
}
```

Card Component:

```
import React from 'react'
import { Card } from 'react-bootstrap'
import { Link } from 'react-router-dom'
function CardComponent({ imageSrc, title, desc, navigationLink, linkText, className }) {
  return (
    <div className='col-md-6 mb-4'>
      <Card className="small-card mb-4">
        <Card.Img variant="top" src={imageSrc} />
        <Card.Body>
          <Card.Title>{title}</Card.Title>
          <Card.Text>
            {desc}
          </Card.Text>
          <Link to={navigationLink} className="btn btn-primary">{linkText}</Link>
        </Card.Body>
      </Card>
    </div>
  )
}

export default CardComponent
```

Complaint Box

Category Component:

```
import CardComponent from './CardComponent';
import categoryData from '../Repository/CategoryData';
function Category() {
  return (
    <div className='container2'>
      <div className='row'>
        {categoryData.map(x => < CardComponent imageSrc={x.imageSrc} title={x.title}
desc={x.desc} navigationLink={x.navigationLink} linkText={x.linkText} />)}
      </div>
    </div>
  );
}
export default Category;
```

Back End Coding :

Functions:

The EduHUB Connect is built on the Model-View-Controller (MVC) architectural pattern, which provides a clear separation of concerns and promotes modularity, maintainability, and scalability. The system architecture consists of three interconnected layers:

1. Model:

- The Model layer represents the application's data and business logic.
- It includes entities such as User, Complaint, AreaInfo, and Admin, encapsulating the data-related operations.
- Business logic related creating new user and complaint processing workflows is implemented in this layer.

2. View:

- User interfaces such as the homepage, login page, userpage, and profile pages are part of the View layer.

3. Controller:

- The Controller layer acts as an intermediary between the Model and View layers, handling user requests and orchestrating the flow of data.
- It contains controller classes that receive incoming HTTP requests, process them, invoke appropriate business logic in the Model layer, and select the appropriate view to render the response.
- Controllers manage the navigation logic, form submissions, and interaction between the user interface and the backend system.

Main Component of Backend:

UserController:

```
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Create([Bind("EmailId")] userEmail
Email)
{
    var existingUser = await _context.Users.FirstOrDefaultAsync(u =>
u.EmailId == Email.EmailId);
    if (existingUser != null)
    {
        return RedirectToAction(nameof(Index));
    }
    User user = new User();
    if (ModelState.IsValid)
    {
        user.UserId = "";
        user.EmailId = Email.EmailId;
        _context.Users.Add(user);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    HttpContext.Session.SetString("EmailId", user.EmailId);
    return Ok(user);
}
```

ComplaintsController:

```
[HttpPost]
[ValidateAntiForgeryToken]
```

Complaint Box

```
public async Task<IActionResult>
Create([Bind("ComplaintType,Description,StreetNo,BuildingNo,PinCode,Victi
mName,VictimAge,VictimGender,Images")] ComplaintInfo complaintInfo)
{
    if (ModelState.IsValid)
    {
        // Create a new Victim_Info entry if victim details are provided
        if (!string.IsNullOrEmpty(complaintInfo.VictimName))
        {
            VictimInfo victimInfo = new VictimInfo
            {
                VictimName = complaintInfo.VictimName,
                VictimAge = complaintInfo.VictimAge,
                VictimGender = complaintInfo.VictimGender
            };

            _context.VictimInfos.Add(victimInfo);
            await _context.SaveChangesAsync(); // Save to trigger the ID
generation
        }

        // Create a new Complaint entry
        Complaint complaint = new Complaint
        {
            ComplaintType = complaintInfo.ComplaintType,
            DateTimeLodged = DateTime.Now,
            StreetNo = complaintInfo.StreetNo,
            BuildingNo = complaintInfo.BuildingNo,
            Description = complaintInfo.Description,
            PinCode = complaintInfo.PinCode,
            Images = complaintInfo.Images
        };

        _context.Complaints.Add(complaint);
        await _context.SaveChangesAsync(); // Save to trigger the ID
generation and other logic in the trigger

        // Create UserComplaint relationship
        var user = await _context.Users.FirstOrDefaultAsync(u =>
u.EmailId == HttpContext.Session.GetString("EmailId"));
        if (user != null)
        {
            UserComplaint userComplaint = new UserComplaint
            {
                UserId = user.UserId,
                ComplaintId = complaint.ComplaintId
            };

            _context.UserComplaints.Add(userComplaint);
            await _context.SaveChangesAsync();
        }

        return RedirectToAction(nameof(Index));
    }
}
```


Complaint Box

```
    }

    // Populate ViewData for dropdowns if ModelState is not valid
    ViewData["AdminAllotedId"] = new SelectList(_context.Admins,
"AdminId", "AdminId", null);
    ViewData["PinCode"] = new SelectList(_context.AreaInfos, "PinCode",
"PinCode", complaintInfo.PinCode);
    ViewData["VictimId"] = new SelectList(_context.VictimInfos,
"VictimId", "VictimId", null);

    return View(complaintInfo);
}
```

AdminController:

```
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Create([Bind("AdminId,AdminPass,AdminStatus")] Admin admin)
{
    if (ModelState.IsValid)
    {
        admin.AdminId = Guid.NewGuid();
        _context.Add(admin);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    return View(admin);
}
```

FUTURE SCOPE

The Complaint Box has some exciting enhancements planned to make it even more useful.

1. To More Areas and concerns: In addition to criminal activity, the platform will cover a broader range of concerns. This means that anyone can utilize it to report a variety of community issues, making it a one-stop shop for all reporting requirements.

2. Providing Expanding Full Tracking: The software will allow users to track their complaints from beginning to end. This will clarify the process and allow consumers to see how their reports are handled until problems are resolved.

3. Adding Multi-Language Support: To ensure that everyone can easily utilize the Complaint Box, multiple languages will be supported. This will allow people from diverse backgrounds to use the platform in their favorites language.

These improvements will make the Complaint Box more flexible, easier to use, and better at helping people keep their communities safe.

CONCLUSION

The Complaint Box is a significant step forward in improving how citizens report problems and engage with law enforcement. Making the procedure easier and more efficient helps to guarantee that concerns are taken seriously and addressed as soon as possible. This project not only addresses present issues in public safety, but also paves the way for future improvements.

In short, the Complaint Box is an essential component of current public safety initiatives. As it progresses, it will make communities safer and more connected. The road to improve public safety is ongoing, and the Complaint Box is poised to play a critical role in assisting residents and law enforcement to collaborate for a safer future.

Bibliography:

We kept window open for further additional development.

- <https://react.dev/>
- <https://www.w3schools.com/REACT/DEFAULT.ASP>
- <https://legacy.reactjs.org/docs/getting-started.html>