

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

Sushant Kumar
REG. NO. 39120124



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
SUSHANT KUMAR (Reg. No: 39120124) who carried out the
project entitled “**isThis-a-Bug: A crowdsourced bug report
system**” under my supervision from February 2022 to April 2022.

Internal Guide
Dr. R.M. Gomathi,

Submitted for Viva voce Examination held on _____

InternalExaminer

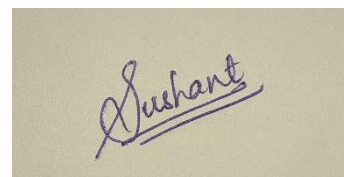
ExternalExaminer

DECLARATION

I, **SUSHANT KUMAR** hereby declare that the project report entitled isThis-a-Bug: A crowdsourced bug report system for large software and product deployment done by me under the guidance of **Dr. R.M. Gomathi** is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Information Technology.

DATE:

PLACE:

A rectangular box containing a handwritten signature in blue ink. The signature appears to be 'Sushant' written in a cursive style.

SIGNATURE OF THE CANDIDATE

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. R.M. Gomathi**, 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

isThis-a-Bug is an approach towards crowdsourced, community driven bug reporting system for large software deployments. The system will include reporting system where the present users of the active deployed software can come forward and report and raise bug tickets from the variety of options, be it UI/UX or performance.

All of the above and many more was enabled by the platform provided by ReactJS, NodeJS, ExpressJS and MongoDB

Deploying large software and maintaining it might be very tough and the same can be said in regards of how the whole bug reporting and patching of experience breaking bugs can be for customers.

isThis-a-Bug is another small step in the same direction to help companies and their customer access to best reporting and best services.

LIST OF FIGURES

| FIGURE NO. | FIGURE NAME | PAGE NO. |
|------------------------|--|------------------|
| <i>Fig 1.1:</i> | <i>MERN stack architecture</i> | <i>4</i> |
| <i>Fig 2.1:</i> | <i>Starting local servers</i> | <i>6</i> |
| <i>Fig 2.2:</i> | <i>DDoS attack check</i> | <i>6</i> |
| <i>Fig 2.3:</i> | <i>Home page/the bug ticketing page</i> | <i>7</i> |
| <i>Fig 2.4:</i> | <i>Sign-up page</i> | <i>7</i> |
| <i>Fig 2.5:</i> | <i>Sign-up confirmation</i> | <i>8</i> |
| <i>Fig 2.6:</i> | <i>Log-in page</i> | <i>8</i> |
| <i>Fig 2.7:</i> | <i>Confirmation after successful report submission</i> | <i>9</i> |
| <i>Fig 2.8:</i> | <i>User past report status console</i> | <i>9</i> |
| <i>Fig 2.9:</i> | <i>About Us page for reference about project and its mission to users</i> | <i>10</i> |

| CHAPTER No. | TITLE | PAGE No |
|--------------------|---|----------------|
| | | i |
| | ABSTRACT | |
| | LIST OF FIGURES | ii |
| 1. | INTRODUCTION | 1 |
| 2. | AIM AND SCOPE OF THE PRESENT INVESTIGATION | 2-3 |
| 3. | MATERIALS AND METHODS USED | |
| | Working with MERN | 4-5 |
| 4. | WALKTHROUGH OF PRODUCT | 6-10 |
| 5. | SUMMARY AND CONCLUSION | 10 |
| 6 | SAMPLE CODE | 11-15 |

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 crowdsourced bug 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

Nobody wants to work with software that doesn't behave as expected. It's a terrible user experience. Bug reporting helps smooth out software, so that it does what it needs to, without frustrating the people using it.

In fact, bug reporting is so important that many development teams have dedicated testers, or Quality Assurance whose job it is to find and report bugs.

Reporting bugs doesn't need to be difficult. Creating a great bug report means that bugs will get fixed faster too. So, what do you actually need for a good bug report?

Here's a 10-step checklist when writing bug reports to make sure you create good bug reports.

Not all the information will need to be included all the time, but it will definitely help you to create a great bug report.

1. **One bug per report.** This is the golden rule of bug reporting. Each bug deserves its own bug report. Why? Because it makes it so much more difficult to resolve bugs when you need to hunt through a bug report that has multiple issues listed. Keep it simple, people!

2. **Where were you in the software when the bug occurred?** Include the URL of the page where the bug is located if it's a web application.

3. **What were you doing in the software when the bug happened?** You may hear this referred to as steps to recreate. When a bug is replicable (able to be recreated) it's easier to work out what went wrong and to fix it.

4. **What were you expecting to happen?** If a bug is software working in an unexpected way, you need to say what you expected it to do.

5. **A screenshot or screen recording of the bug.** If a picture is worth 100 words, including a screenshot or screen recording of the bug in your bug report may prove easier than trying to explain what you're experiencing.
6. **Record technical information.** Include information such as your operating system, what browser you are using for websites and web applications. Include whether you were using a desktop or mobile, if it's not already clear.
7. **Include any error messages and codes you received.** If you're getting a specific error message or code, it will be helpful when pinpointing what the bug is and how to resolve it. Note, unhelpful error messages and codes can be minor bugs themselves.
8. **Can you replicate the issue?** Does the same thing happen every time you try to complete the task you're doing? Including this information in bug reports will help the developer find the cause of the bug.
9. **Have you tried to fix it yourself?** Look, there's a reason IT always asks "Have you tried turning it off and on again?" or whether you've tried refreshing the webpage you're on. Include whether you've tried these simple tests in your bug report. It'll save time on follow up emails or calls.
10. **How much is it impacting your work?** How much a bug affects your ability to complete the task you want determines the order in which bugs are resolved. You may hear the term priority or severity used to determine how important bugs are and how quickly they need to be fixed. Typically, severity varies from Very High (it stops you from working completely) down to Very Low (cosmetic changes). Including this information helps development teams prioritize the order they resolve bugs.

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

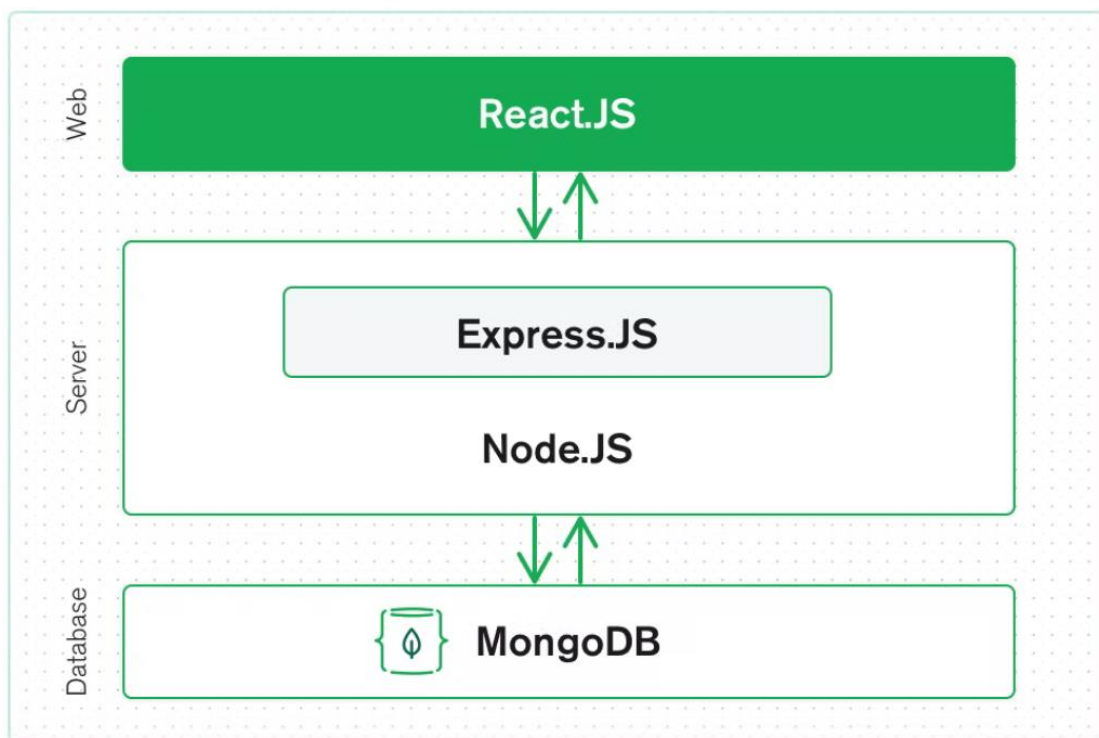


Fig 1.1: MERN stack architecture

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.

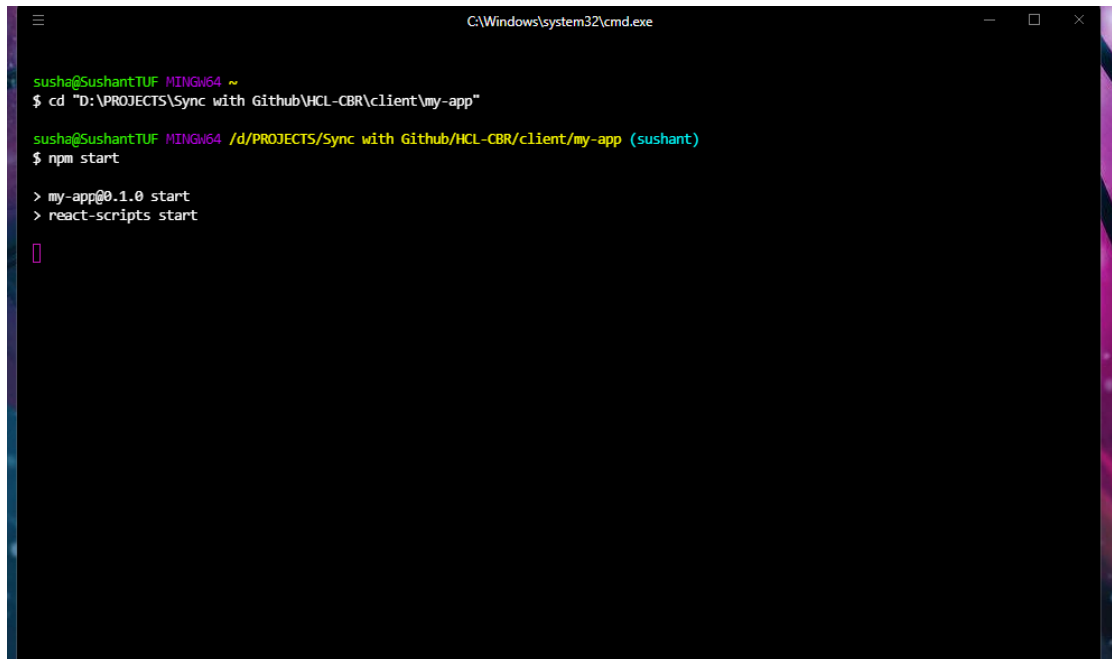
MongoDB Database Tier

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

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



```
C:\Windows\system32\cmd.exe

susha@sushantTUF MINGW64 ~
$ cd "D:\PROJECTS\Sync with Github\HCL-CBR\client\my-app"

susha@sushantTUF MINGW64 /d/PROJECTS/Sync with Github/HCL-CBR/client/my-app (sushant)
$ npm start

> my-app@0.1.0 start
> react-scripts start

█
```

Fig 2.1: Starting local servers

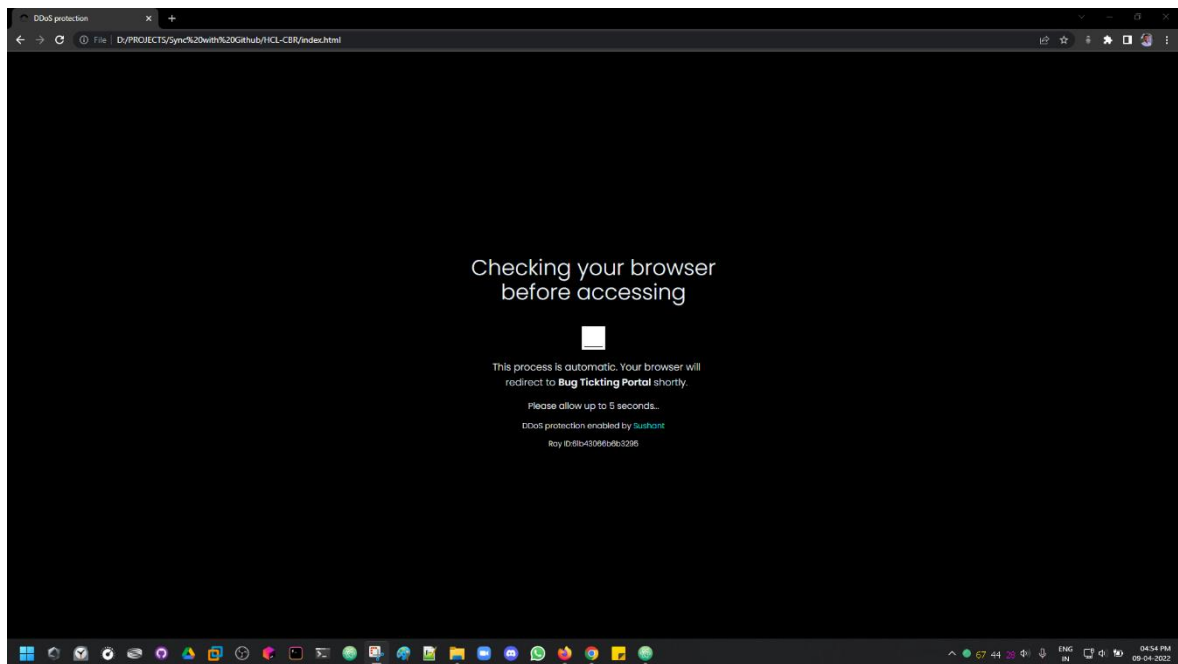


Fig 2.2: DDoS attack check

The main page and the home page of the portal

The screenshot shows a web browser window with the URL `D:\PROJECTS\Sync\20with\20Githu\HCL_CBR\Experimental\features\demoBuild\BugReportPage\index.html`. The page has a dark theme and a navigation bar with links: **isThis-a-Bug**, **HOME**, **ABOUT**, **FAQ**, and **LOGIN**. The main content area is titled **RAISE YOUR BUG REPORT** and contains a form with the following fields:

- Defect Reporting Date**: A date picker showing `dd-mm-yyyy`.
- Time of Event**: A time picker showing `HH:MM:SS`.
- Title/Summary**: A text input field with a placeholder "Keep under 20 words..".
- Your Name**: A text input field with a placeholder "Enter your name here".
- Project Name**: A dropdown menu with the option `--Please choose an option--`.
- Release/Build version**: A dropdown menu with the option `--Please choose an option--`.
- Website URL**: A text input field with a placeholder "URL of the illegal trade website".
- How Detected**: A dropdown menu with the option `--Please choose an option--`.
- Defect/Enhancement**: A dropdown menu with the option `--Please choose an option--`.

The Windows taskbar at the bottom shows the time as 04:56 PM on 09-04-2022.

Fig 2.3: Home page/the bug ticketing page

> The main page above is where all the stuff happens. At this page the user can use the given digital form to fill the form so as to raise a successful bug report (given that they already have an account and are currently signed in). If not so then the user will then be prompted to sign up in case where they don't have an account or just made to log to have a successful bug report raised.

The screenshot shows a web browser window with the URL `D:\PROJECTS\Sync\20with\20Githu\HCL_CBR\Experimental\features\demoBuild\AdminSignUp\index.html`. The page has a dark theme and a navigation bar with links: **isThis-a-Bug**, **HOME**, **ABOUT**, **FAQ**, and **LOGIN**. The main content area is titled **ADMIN SIGN UP** and contains a form with the following fields:

- Name**: A text input field with a placeholder "Enter your Name".
- Unique Username**: A text input field with a placeholder "Try entering a Unique Username".
- Email**: A text input field with a placeholder "Enter your e-mail here".
- Password**: Two text input fields, one with a placeholder "Try entering a secure password" and another with a placeholder "Re-enter your password".
- Unique Admin Code**: A text input field with a placeholder "Enter the secret code provided by it".

At the bottom of the form is a **Sign Up** button. The Windows taskbar at the bottom shows the time as 05:03 PM on 09-04-2022.

Fig 2.4: Sign-up page

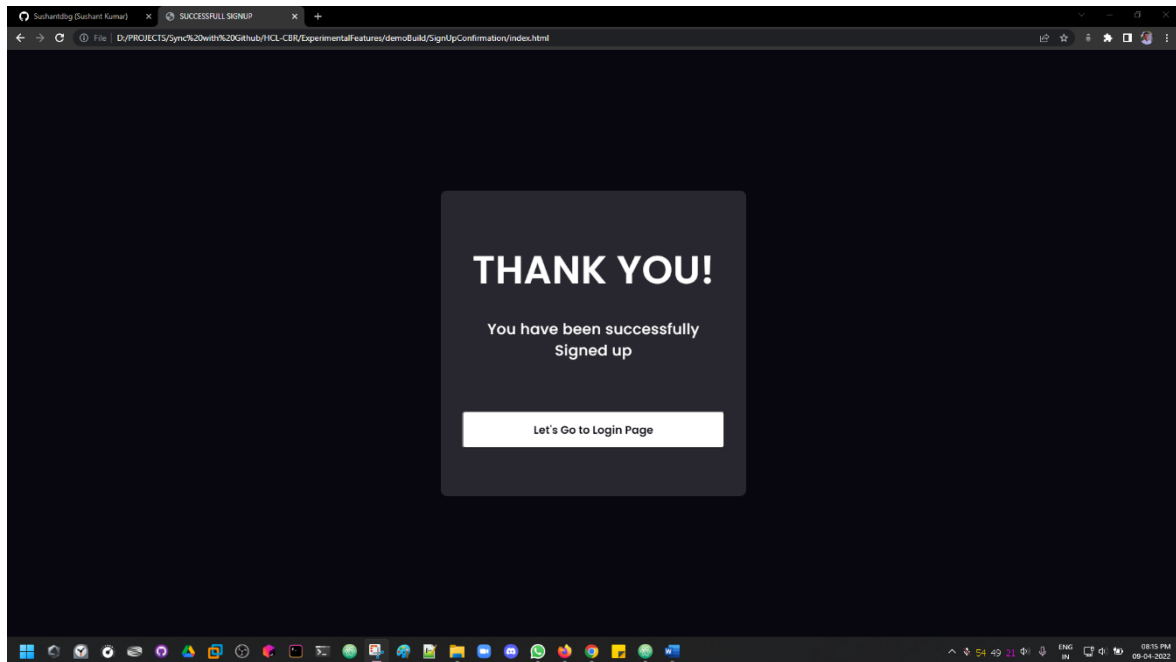


Fig 2.5: Sign-up confirmation

> After successful account creation one can carry on with the login and later submit the bug report

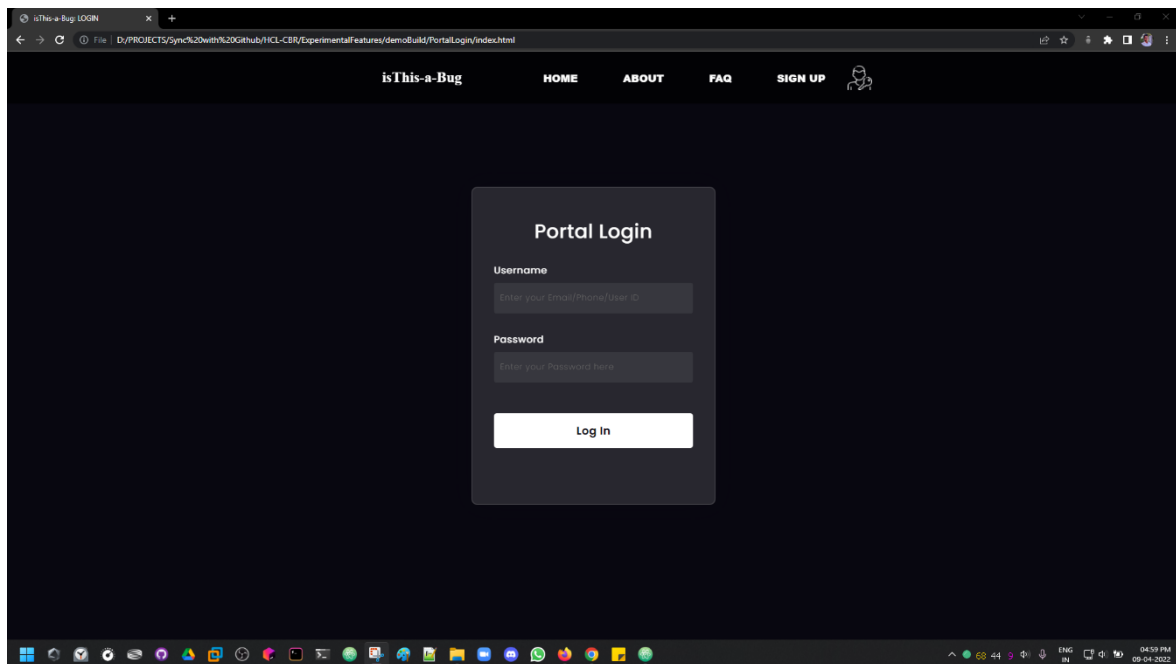


Fig 2.6: Log-in page

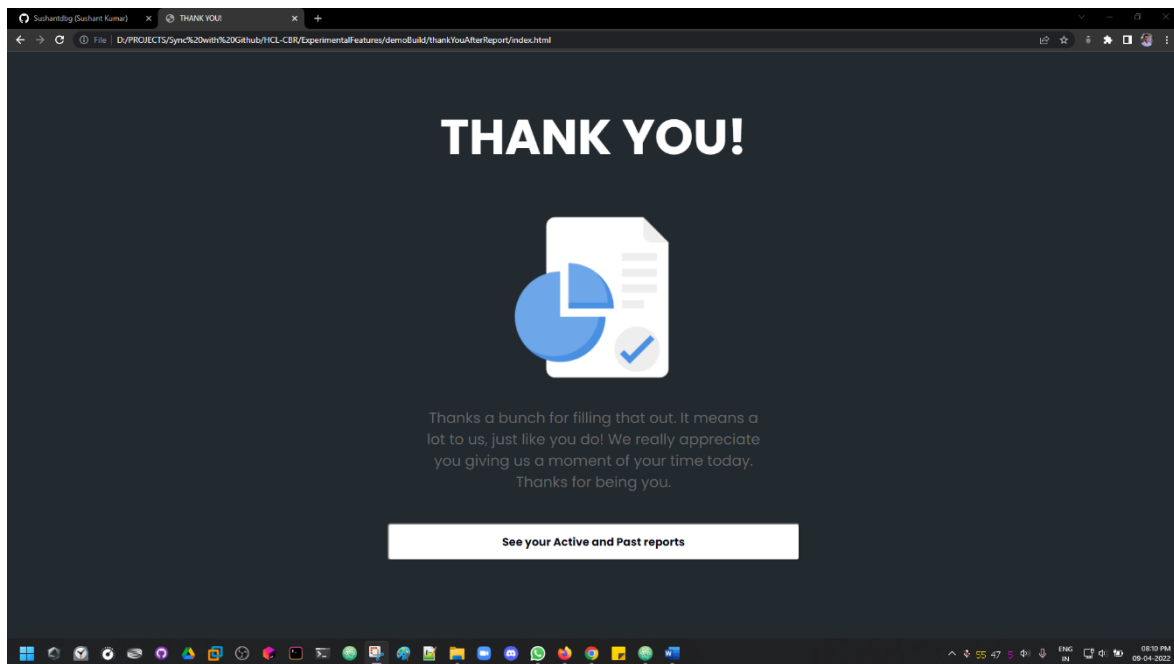


Fig 2.7: Confirmation after successful report submission

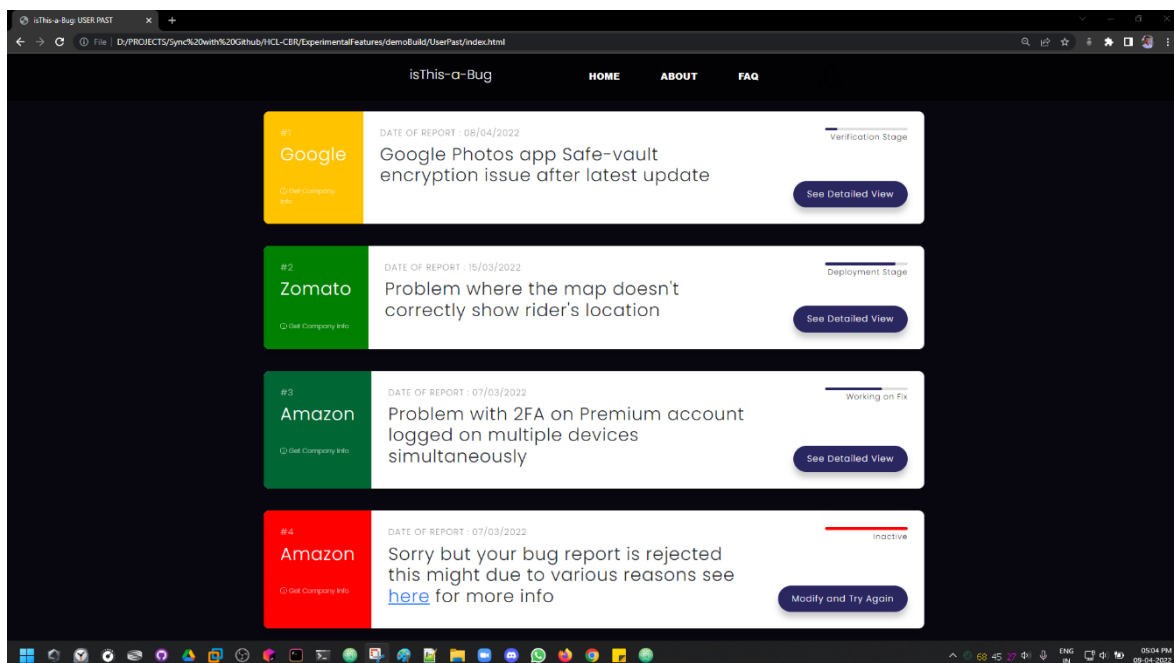


Fig 2.8: User past report status console

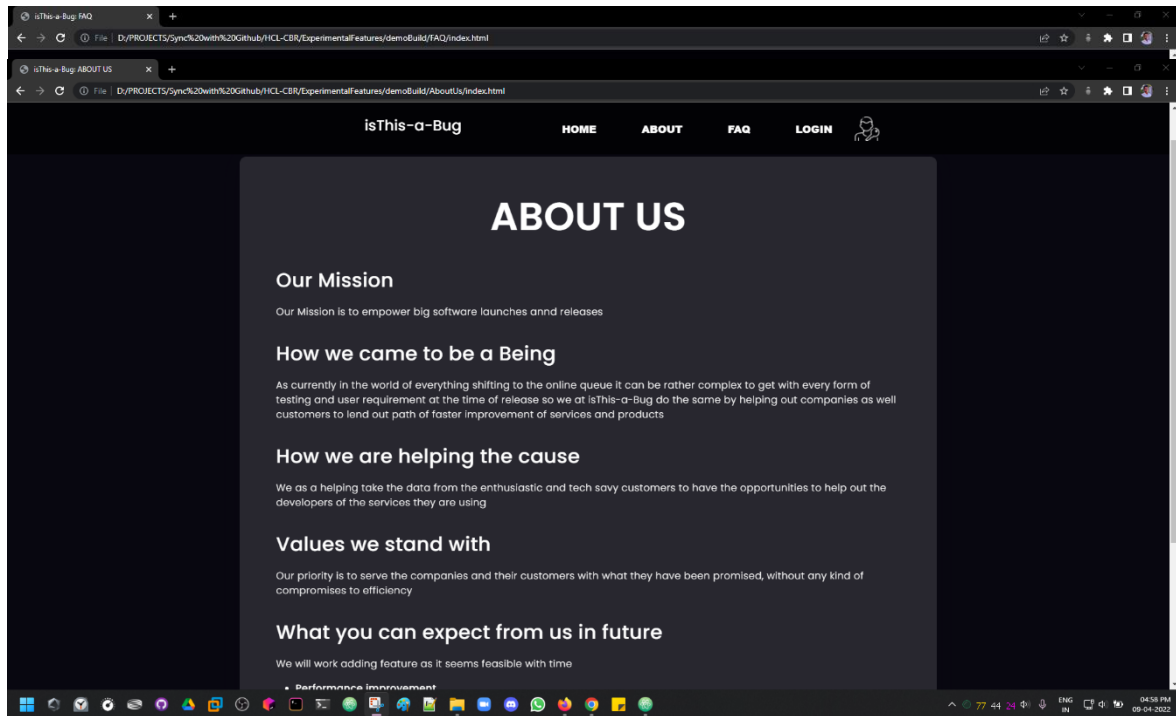


Fig 2.9: About Us page for reference about project and its mission to users

CHAPTER 5: SUMMARY AND CONCLUSION

There are numerous architectural and conceptual considerations when you are contemplating a working bug reporting system model for any major application deployment. As clear by now the benefits of having an effective, highly scalable bug reporting infrastructure that grows with your product, isn't just limited to bug fixes but to the edge customer satisfaction and experience; A link and path of transparency between the users and the developers so as to learn maintain and grow together.

SAMPLE CODE (Main Page Form)

```
import React from 'react';
import ReactDOM from 'react-dom';

export default function MainPageForm(){
  return(

    <div class="main--form--page">

      <form class="mainpage--form" action="index.html" method="post">
        <h2 class="mainpage--title">RAISE YOUR BUG REPORT</h2>
        <div class="mainpage--date--div">
          <div class="mainpage--date--form">
            <label class="mp--all--labels" for="dateOfEvent"><span class="">Defect
            Reporting Date</span></label>
            <input type="date" name="" value="" class="mp--input--field form-
            control date-input">
          </div>
        </div>

        <div class="div-time">
          <div class="time-form">
            <label class="mp--all--labels" for="" class="time-label"><span
            class="">Time of Event</span></label>
            <input type="time" name="" value="" class="mp--input--field time-input
            form-control">
          </div>
        </div>

        <div>
          <label class="mp--all--labels" for="fname" class="accu-
          name">Title/Summary</label>
          <input type="text" id="fname" name="firstname" placeholder="Keep
          under 20 words..." class="mp--input--field guilty-name form-control">
        </div>

        <div>
          <label class="mp--all--labels" for="fname" class="accu-name">Your
          Name</label>
          <input type="text" id="fname" name="firstname" placeholder="Enter
          your name here" class="mp--input--field guilty-name form-control">
        </div>

        <div class="drug-rel">
```

```
<label class="mp--all--labels" for="" class="custom-select">Project  
Name</label>
```

```
<select class="form-control" class="rep-select">  
<option class="mp-option-all" value="0">--Please choose an option--  
</option>  
<option class="mp-option-all" value="1">Fedora 35</option>  
<option class="mp-option-all" value="2">Zomato</option>  
</select>  
</div>
```

```
<div class="drug-rel">  
<label class="mp--all--labels" for="" class="sever"> Release/Build  
version</label>  
<select class="form-control">  
<option class="mp-option-all" value="">--Please choose an option--  
</option>  
<option class="mp-option-all">Current Live Production Build</option>  
<option class="mp-option-all">Beta Build</option>  
<option class="mp-option-all">Alpha Build</option>  
</select>  
</div>
```

```
<div>  
<label class="mp--all--labels" for="fname" class="accu-name">Website  
URL</label>  
<input type="text" id="fname" name="firstname" placeholder="URL of  
the illegal trade website" class="mp--input--field guilty-name form-  
control">  
</div>
```

```
<div class="drug-rel">  
<label class="mp--all--labels" for="" class="custom-select">How  
Detected</label>  
<select class="form-control" class="rep-select">  
<option class="mp-option-all" value="0">--Please choose an option--  
</option>  
<option class="mp-option-all" value="1">Testing</option>  
<option class="mp-option-all" value="2">Review</option>  
<option class="mp-option-all" value="3">Walkthrough</option>  
</select>  
</div>
```

```

<div class="drug-rel">
<label class="mp--all--labels" for=""
class="sever">Defect/Enhancement</label>
<select class="form-control">
<option class="mp-option-all" value="">--Please choose an option--
</option>
<option class="mp-option-all">Defect</option>
<option class="mp-option-all">Enhancement</option>
</select>
</div>

```

```

<div class="drug-rel">
<label class="mp--all--labels" for="" class="custom-
select">Environment</label>
<select class="form-control" class="rep-select">
<option class="mp-option-all" value="0">--Please choose an option--
</option>
<option class="mp-option-all" value="1">Windows</option>
<option class="mp-option-all" value="2">Linux</option>
<option class="mp-option-all" value="3">MacOS</option>
<option class="mp-option-all" value="4">Android</option>
<option class="mp-option-all" value="4">iOs</option>
<option class="mp-option-all" value="5">Multiple</option>
</select>
</div>

```

```

<div class="drug-rel">
<label class="mp--all--labels" for="" class="sever">Priority</label>
<select class="form-control">
<option class="mp-option-all" value="">--Please choose an option--
</option>
<option class="mp-option-all">Mild</option>
<option class="mp-option-all">Priority</option>
<option class="mp-option-all">High Priority</option>
<option class="mp-option-all">Extreme</option>
</select>
</div>

```

```

<div class="drug-rel">
<label class="mp--all--labels" for="" class="custom-
select">Severity</label>
<select class="form-control" class="rep-select">
<option class="mp-option-all" value="0">--Please choose an option--
</option>

```

```

<option class="mp-option-all">Okk</option>
<option class="mp-option-all">Bad</option>
<option class="mp-option-all">Breaking</option>
<option class="mp-option-all">Extreme</option>
</select>
</div>

```

```

<div class="drug-rel">
<label class="mp--all--labels" for="" class="sever">Status</label>
<select class="form-control">
<option class="mp-option-all" value="">--Please choose an option--
</option>
<option class="mp-option-all">Present</option>
<option class="mp-option-all">Patched</option>
</select>
</div>

```

```

<div class="addr-div">
<label class="mp--all--labels" for="" class="form-label" class="addr-
label">Enter detailed description</label>
<textarea class="addr-text" name="textarea" rows="8" cols="80"
id="inputAddress" placeholder="Enter Drug Activity related
address"></textarea>
</div>

```

```

<div class="addr-div">
<label class="mp--all--labels" for="" class="form-label" class="addr-
label">Steps to Reproduce</label>
<textarea class="addr-text" name="textarea" rows="8" cols="80"
id="inputAddress" placeholder="Enter Drug Activity related
address"></textarea>
</div>

```

```

<div class="input-group">
<label class="mp--all--labels" class="input-group-text"
for="inputGroupFile01"> Upload relevant Screenshots </label>
<input type="file" class="mp--input--field form-control add-file"
id="inputGroupFile01" placeholder="">
</div>

```

```

<div class="col-4">
  <div class="form-check">
    <input class="mp--input--field form-check-input" type="checkbox"
    id="gridCheck">
    <label class="mp--all--labels" class="form-check-label" for="gridCheck">
      I have checked data for any errors
    </label>
  </div>
</div>

  <div class="main-button">
    <button type="button"
    onclick="window.location.href='../thankYouPage/index.html'" class="btn
    btn-primary mp-button-last">SUBMIT</button>
  </div>

</form>

</div>

)
}

```