**INTERDISCIPLINARY PROJECT REPORT**

**at**

**Sathyabama Institute Of Science And Technology**

**(Deemed to be University)**

Submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering

By

**ISHANT SINGH**

**(Reg No.40110465)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

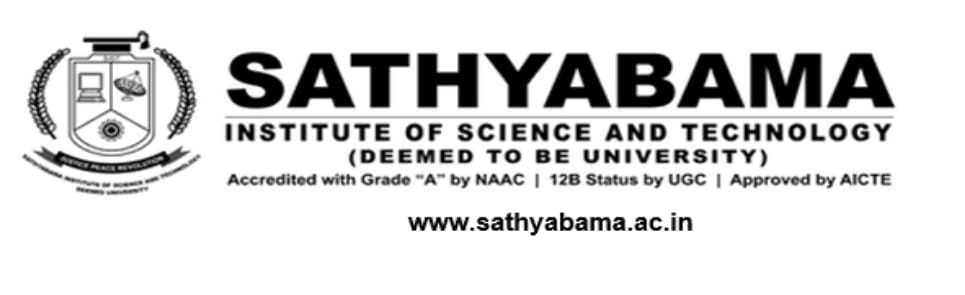
**SCHOOL OF COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI**

**CHENNAI – 600119, TAMILNADU.**

**April 2023**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the Bonafide work of **ISHANT SINGH** (**Reg No: 40110465**) who carried out the project entitled “**QR CODE GENERATOR**” under mysupervision from JAN 2023 to APR 2023.

**Internal Guide**

**Dr.D. Usha Nandini**

**Head of the Department**

**Dr. L. Lakshmanan M.E., Ph.D.**

**Submitted for Viva voice Examination held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Internal Examiner External Examiner**

**DECLARATION**

I **ISHANT SINGH** hereby declare that the Project Report entitled “**QR CODE GENERATOR**“ done by me under the guidance of **Dr.D. Usha Nandini** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

**DATE:**

**PLACE:**

**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. L. Lakshmanan M.E., Ph.D.,** Head of the Department of Computer Science and Engineering 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. Usha Nandini**, 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 Computer Science and Engineering** who were helpful in many ways for the completion of the project.

**TRAINING CERTIFICATE**

# 

**ABSTRACT**

We have shown the importance of URL optimizing at the time of sharing on various platforms. URLs appear to be long, unattractive, on most of the social platforms. They often get broken when shared on social media platforms like e-mail and short URLs can come in handy during these times. To shorten an Internet Long URL, we proposed a URL shortening Web service that will take a long Web URL/address and creates a shorter URL that will not break when we share on different platforms and even Generating a QR Code for the Link Shortened. Websites with short URL will be helpful for the companies to expand their business in many ways URL Shortener helps the client to remember the URL very easily and use it easily the generator works by converting the user's input into a matrix of black and white squares, which can then be read by a QR code scanner. This abstract provides an overview of QR code generators and their applications, as well as the technology behind them. This project aims to build a QR code generator using ReactJS, a popular front-end JavaScript library. The QR code generator allows users to generate custom QR codes for various purposes, including marketing, authentication, and information sharing. The application is built using ReactJS and other open-source libraries such as react-qr-code and react-copy-to-clipboard. It is designed to be user-friendly and customizable, with various options for QR code size, color, and content. Users can also download and save the generated QR codes for later use. The application utilizes modern web development practices such as responsive design and accessibility to ensure that the QR code generator is accessible to all users regardless of their device or ability. Overall, this project aims to provide a simple yet powerful QR code generator using ReactJS and other modern web technologies.

|  |  |  |
| --- | --- | --- |
| **TABLE OF CONTENTS** | | |
| **CHAPTER No.** | **TITLE** | **PAGE No** |
|  | **ABSTRACT** | vi |
|  | **LIST OF FIGURES** | viii |
| 1 | **INTRODUCTION** | 1 |
|  | 1.1 WEB DEVELOPMENT | 1 |
|  | 1.2 REACT.JS | 2 |
|  | 1.2.1 FEATURES OF REACT.JS | 3 |
|  | 1.3 LITERATURE SURVEY | 4 |
| **2** | **AIM AND SCOPE OF THE PRESENT INVESTIGATION** | 4 |
|  | 2.1 AIM | 6 |
|  | 2.2 SCOPE | 7 |
|  | 2.3 PROBLEM STATEMENT | 8 |
| **3** | **EXPERIMENTAL OR MATERIALS AND METHODS; ALGORITHMS USED** | 9 |
|  | 3.1 SYSTEM REQUIREMENTS | 10 |
|  | 3.1.1 HARDWARE REQUIREMENTS | 11 |
|  | 3.1.2 SOFTWARE REQUIREMENTS | 11 |
|  | 3.2 INSTALLATION AND SETUP | 12 |
|  | 3.2.1 INSTALLING VS CODE | 12 |
|  | 3.3 MODULES | 12 |
|  | 3.3.1 FRONT END | 13 |
|  | 3.3.2 BACK END | 13 |
|  | 3.4 ARCHITECTURE DIAGRAM | 14 |
| **4** | **RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS** | 16 |
|  | 4.1 REACT APP | 17 |
|  | 4.2 QR CODE INPUT | 18 |
|  | 4.3 QR CODE OUTPUT | 19 |
|  | 4.4 LINK GENERATION | 20 |
| **5** | **SUMMARY AND CONCLUSIONS** | 22 |
|  | **REFERENCES** | 23 |
|  | **APPENDIX** |  |
|  | **FUTURE WORK** | 24 |
|  | A.SCREENSHOTS | 25 |
|  | B.SOURCE CODE | 26 |

|  |  |  |
| --- | --- | --- |
| **LIST OF FIGURES** | | |
| **FIGURE NO.** | **FIGURE NAME** | **PAGE NO.** |
| 1.1 | REACT.JS | 3 |
| 3.1 | npm START | 11 |
| 3.2 | INSTALLED GIT | 12 |
| 3.3 | React-qrcode IN BROWESER | 13 |
| 3.4 | ARCHITECTURE DIAGRAM | 14 |
| 3.5 | ARCHITECTURE DIAGRAM | 15 |
| 4.1 | REACT APP | 17 |
| 4.2 | QR CODE INPUT | 19 |
| 4.3 | QR CODE OUTPUT | 20 |
| 4.4 | LINK GENERATION | 21 |

**CHAPTER 1**

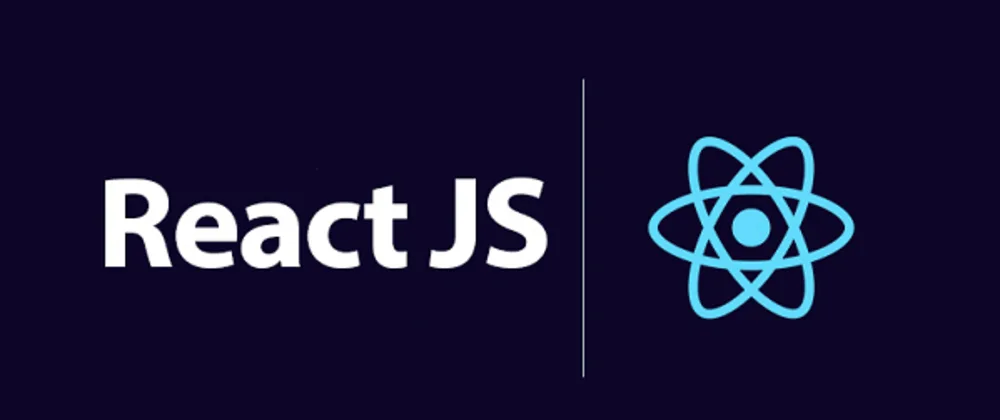
**INTRODUCTION**

**1.1 WEB DEVELOPMENT**

Web development is the process of creating websites. It includes everything from planning and designing to coding and publishing. Web developers use a variety of technologies to create a website. The most common ones are HTML, CSS, and JavaScript. HTML (Hypertext Markup Language) is used to structure the content of a website. CSS (Cascading Style Sheets) is used to style the website's content. JavaScript is used to add interactivity to a website. Web development is a collaborative process. A team of designers, developers, and testers works together to create a website. The process starts with a plan. The team decides what the website will be about and what features it will have. Then, the designers create the website's layout and design. The developers write the code that makes the website work. The testers make sure the website is working properly. Finally, the website is published and made available to the public. Web development is a constantly evolving field. New technologies are being created all the time. This means that there are always new things to learn. If you're interested in a career in web development, you need to be prepared to stay up-to-date with the latest trends. The World Wide Web (WWW) is a network of online content that is formatted in HTML and accessed via HTTP. The term "web development" refers to the design and development of websites that are hosted on the WWW. A web developer is a software engineer who specializes in the development of websites. The early days of the WWW were characterized by a lack of standardization and a Wild West atmosphere. Websites were developed using a variety of programming languages and web development tools. This led to a situation where there was a lot of inconsistency in the way that websites looked and operated. The introduction of web standards by the World Wide Web Consortium (W3C) in the late 1990s began to change the landscape of web development. The W3C standards provided a common set of guidelines for the development of websites. These standards helped to ensure that websites were more consistent in their design and function. The modern-day web is a far cry from the early days of the WWW. Web development has come a long way in a short amount of time. The introduction of web standards has helped to create a more consistent and accessible web. And the proliferation of powerful web development tools has made it easier than ever for developers to create complex and dynamic websites.

**1.2 REACT.JS:**

React.js is a JavaScript library for building user interfaces. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications. The main purpose of React is to be fast, scalable, and simple. It works by breaking down complex UI into small, reusable pieces. This makes it easy to create and maintain complex UIs. React is also used by Facebook in their products, such as the Facebook website, the Facebook app, and Instagram. React is open source, so anyone can use it to create their own applications. React is a great choice for building user interfaces. It is fast, scalable, and simple to use. If you are looking for a JavaScript library to help you build complex UIs, React is a good option.



***Fig. 1.1: React.js***

* + 1. ***Features of React.js***

**Some features of React.js include:**

* React is a JavaScript library for building user interfaces
* It is declarative, meaning that you can simply describe what your UI should look like, without having to write code to specifically make it happen
* It is component-based, meaning that your UI is composed of small, reusable pieces called "components"
* It is fast, meaning that it can efficiently update your UI in response to user input
* It is flexible, meaning that you can use it in conjunction with other libraries and frameworks

**1.3 LITERATURE SURVEY**

QR code technology has been increasingly popular in recent years as a powerful tool for information sharing and marketing. A literature survey on QR code generator reveals various research and development efforts in this field.

One study proposed a QR code generator using image processing techniques to ensure the readability and reliability of generated codes. The proposed system achieved high accuracy and performance in generating QR codes with error correction capability.

Another study focused on the optimization of QR code generation for mobile devices. The study presented a lightweight QR code generator that utilized the mobile device's hardware resources to improve the generation speed and reduce the processing load.

Some researchers have explored the use of QR codes in healthcare applications. One study proposed a QR code generator for patient identification that could be used in emergency medical situations. The generator could quickly generate QR codes containing the patient's medical history and other vital information, improving the speed and accuracy of patient care.

In addition to research, many developers have created open-source QR code generators for various programming languages and platforms. For example, the ZXing library is a popular open-source QR code generator for Java, while qrcode.js is a JavaScript-based QR code generator for web applications.

Overall, the literature survey shows that QR code generator technology has been extensively researched and developed. The research and development efforts have focused on improving the reliability, speed, and accuracy of QR code generation and exploring new applications of QR codes in various fields.

**CHAPTER 2**

**AIM AND SCOPE OF THE PRESENT INVESTIGATION**

**2.1 AIM OF THE PROJECT:**

The pivotal aim of this project is to develop a web server to Shorten the URL’s and generate the QR Code.

Here we use this technique to shorten the long URL Link’s which make easy to memorize or send and to create a QR Code to make more easily to access Link’s just by scanning.

We use this model to help Users to access Links as easier and as fast as possible and easy to share with multiple number of User’s simultaneously.

**2.2 SCOPE OF THE PROJECT:**

The scope of the application is very much in advance. This Web application can be used for many purposes:

Communication

Collaboration

Business Growth

Marketing

It ensures that you get the right messages out to your audience without taking up too much space in your social posts.

Thus the scope of this project is optimize the Valid URL for the User Using Shortening and QR Technique.

**2.3 PROBLEM STATEMENT**

Create a URL shortener library (e.g., bitly). Given a long URL, your program should return a shortened URL. Ensure the following:

1. For a long URL X, the program should always return the shortened URL x.

2. The short URLs the program returns, must not follow any pattern successive calls to the program should return very different short URLs.

3. For two different long URLs X and Y, the program should (ideally) always return two different QR code.

4. The program must be able to accept long URLs provided both through

**CHAPTER 3**

**MATERIAL AND METHODS USED**

**3.1 HARDWARE AND SOFTWARE REQUIREMENTS:**

***3.1.1 Hardware Requirements***

* Processor: 5th gen Intel core (i3)
* Clock-speed: 1.7GHz
* Memory Space: 500GB
* RAM: 4GB
* Display: 1024 x 768
* Mouse: Digital
* Keyboard: 105 Digital keys

***3.1.2 Software Requirements***

* Operating System: Windows
* VSCode: To handle HTML5, Bootstrap, JavaScript code

**3.2 INSTALLATION AND SETUP**

***3.2.1 Installing VS Code on Windows***

**Step 1**. Download VS Code for Windows from the official website.

**Step 2**. Run the installer, and follow the on-screen instructions.

**Step 3**. Once the installation is complete, launch VS Code.

**Step 4**. Follow the prompts to sign in with your Microsoft account, and agree

to the terms of service.

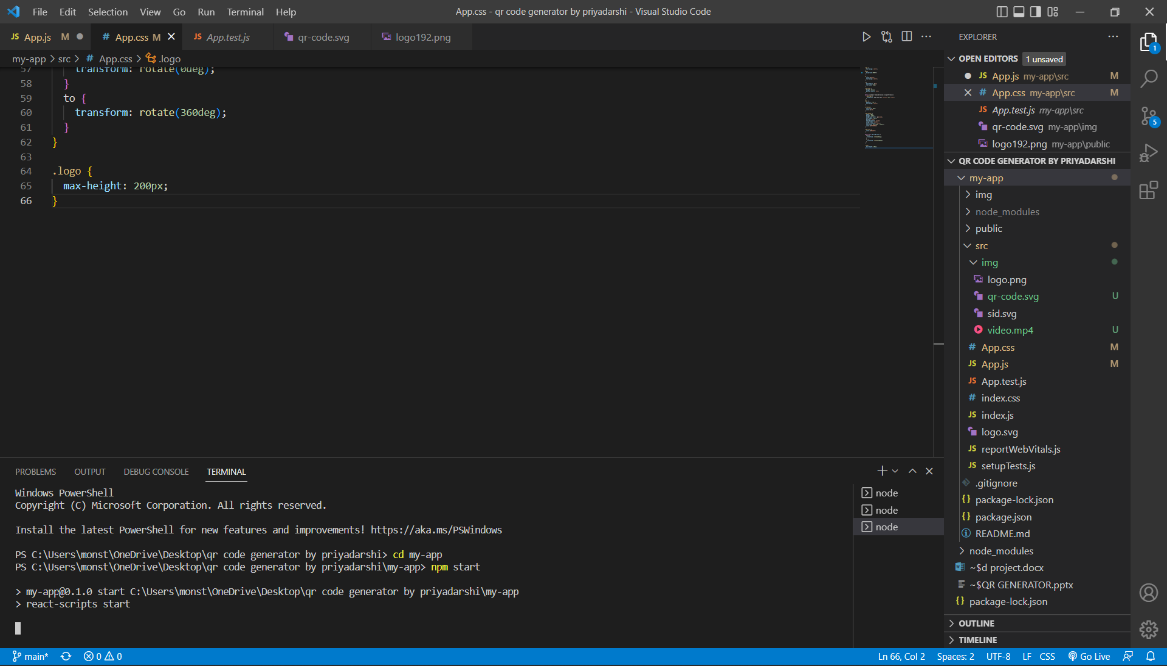
**Step 5**. Once signed in, you'll be able to use all of VS Code's features.

**3.3 MODULES**

***3.3.1 FRONT-END***

**Step 1.** User will open the React.JS framework.

**Step 2.** User will create the my-app.

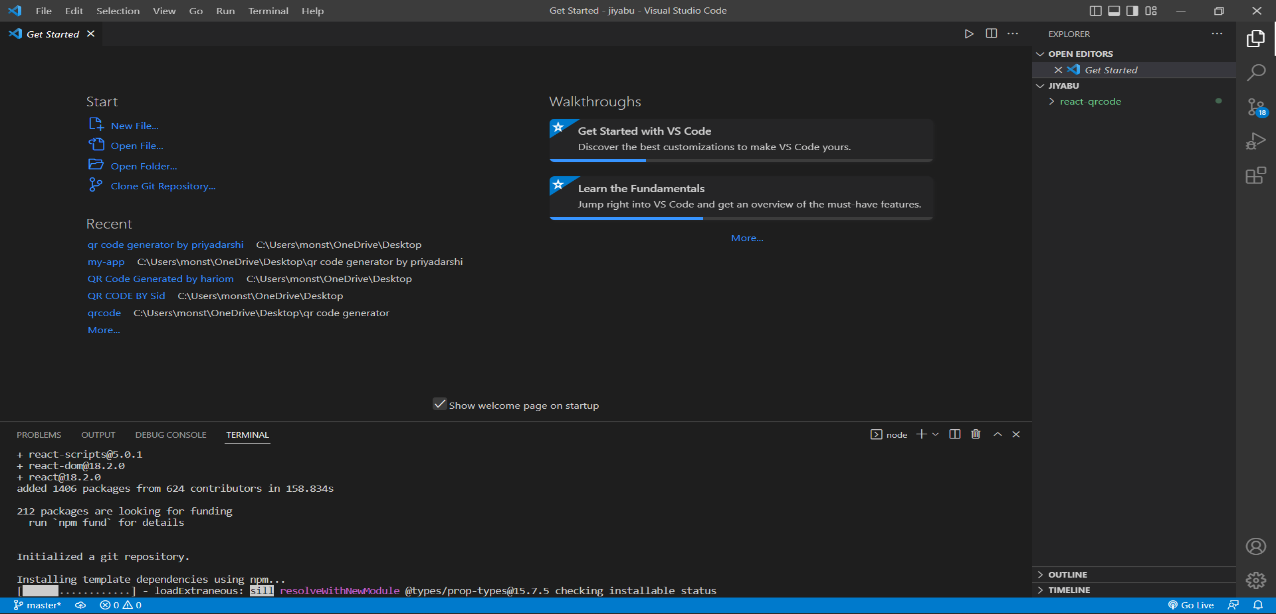
**Step 3.** User will start the npm package.

***Fig 3.1 npm start***

**Step 4.** User will get the QR Code and would be able to enter the link.

***3.3.2 BACK-END***

**Step 1.** You will need Git and Node.js installed on your computer.

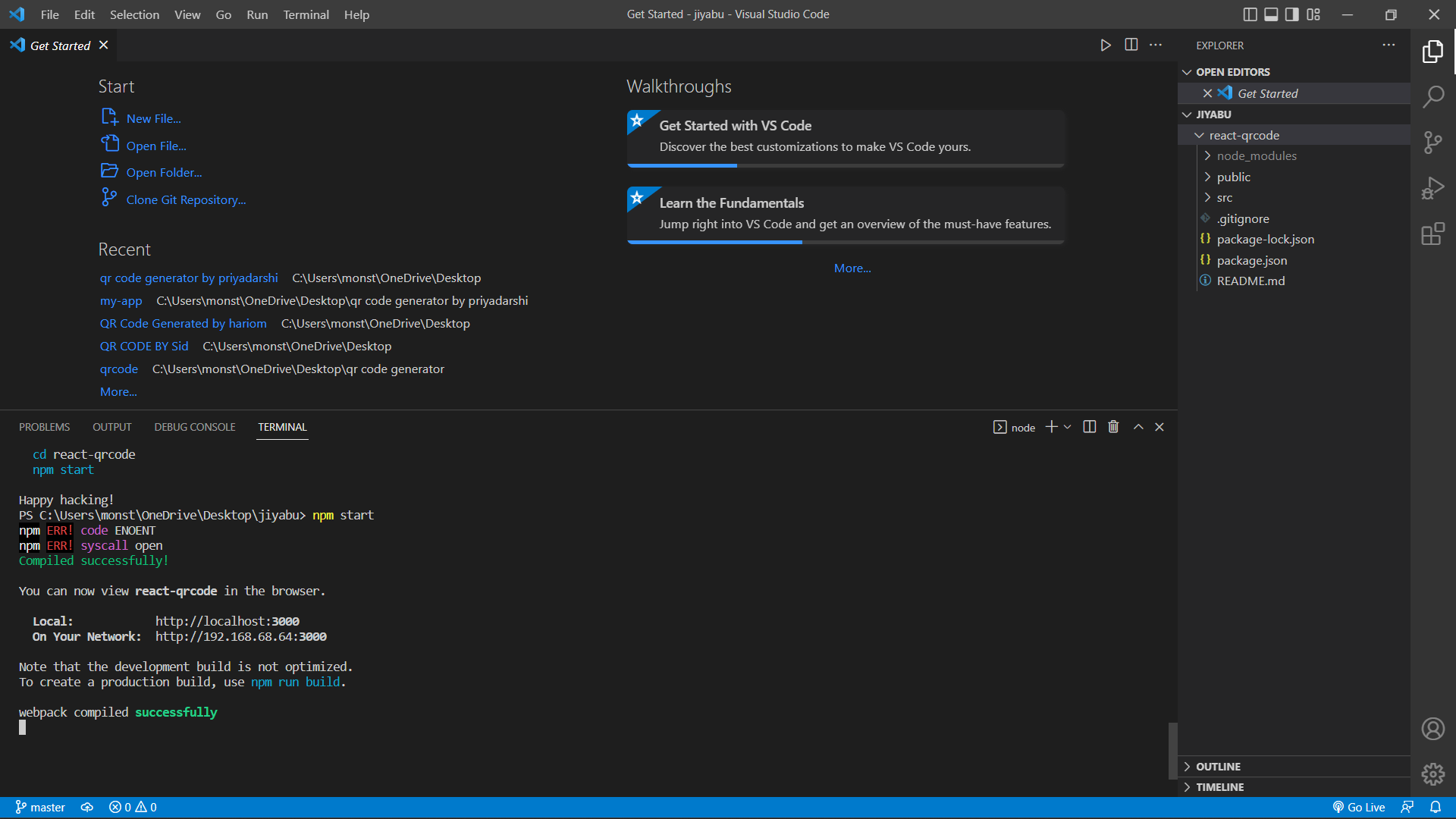


***Fig 3.2 installed GIT repository***

**Step 2.** To install node/install dependencies: npm install (in console/terminal).

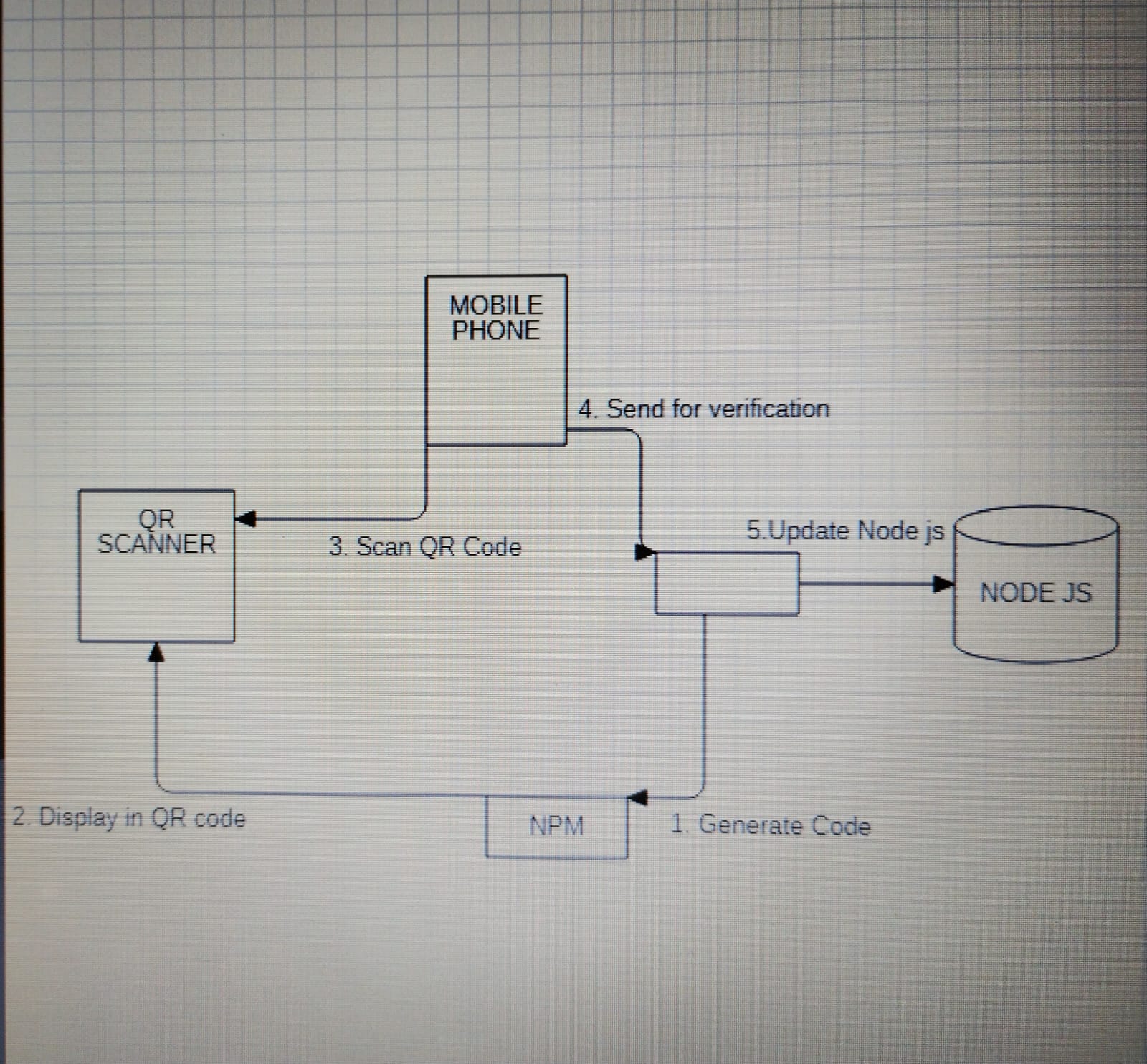
**Step 3.** npm install node-fetch, npm start runs the app in the development mode**.**

**Step 4.**  npx package which is Node package execute which will take the package and give to the Node package manager(npm) and then those will be used in the front end .



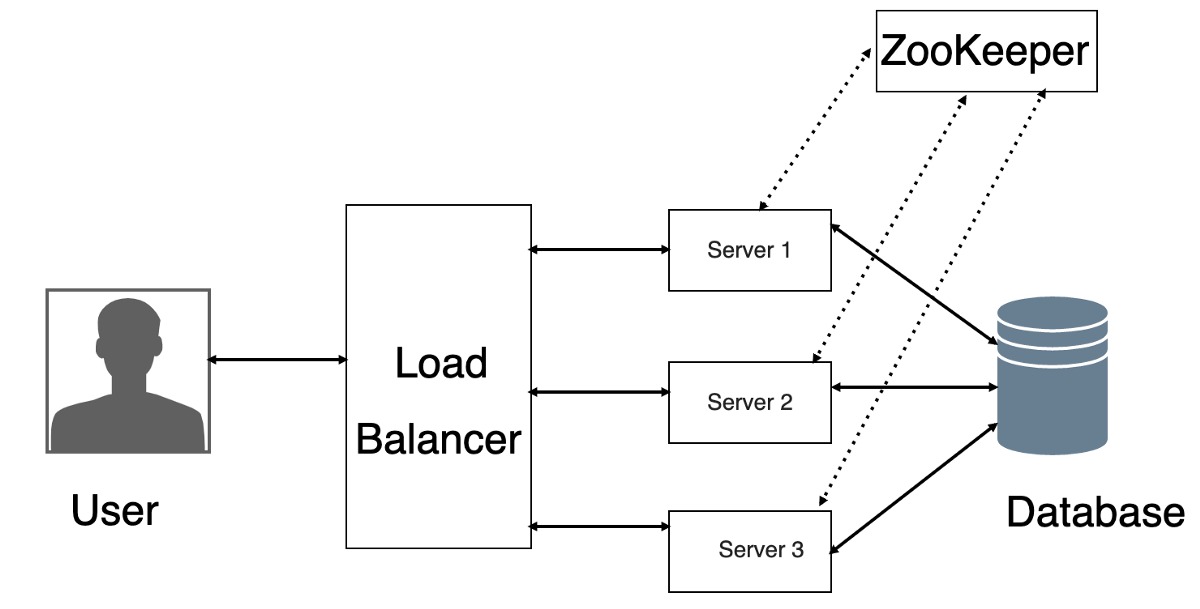
***Fig 3.3 react-qrcode in browser***

**3.4 ARCHITECTURE DIAGRAM**

****

***Fig 3.4 QR Code Architecture***

The architecture diagram for a QR code generator using ReactJS consists of three main components: the form component, the QR code component, and the main component that manages the state of the application. The form component is responsible for capturing user input, which is then passed to the main component as a prop. The main component manages the state of the application, including the value of the input and the QR code data. When the user clicks the "Generate QR Code" button in the form, the main component updates its state with the input value, and the QR code component is rendered with the new data. The QR code component is responsible for rendering the QR code using the “qrcode.react” package. This architecture allows for a clean separation of concerns between the form component and the QR code component, and allows for easy management of state in the main component.



***Fig 3.5 Backend Architecture***

**CHAPTER 4**

**RESULTS AND PERFOMANCE ANALYSIS**

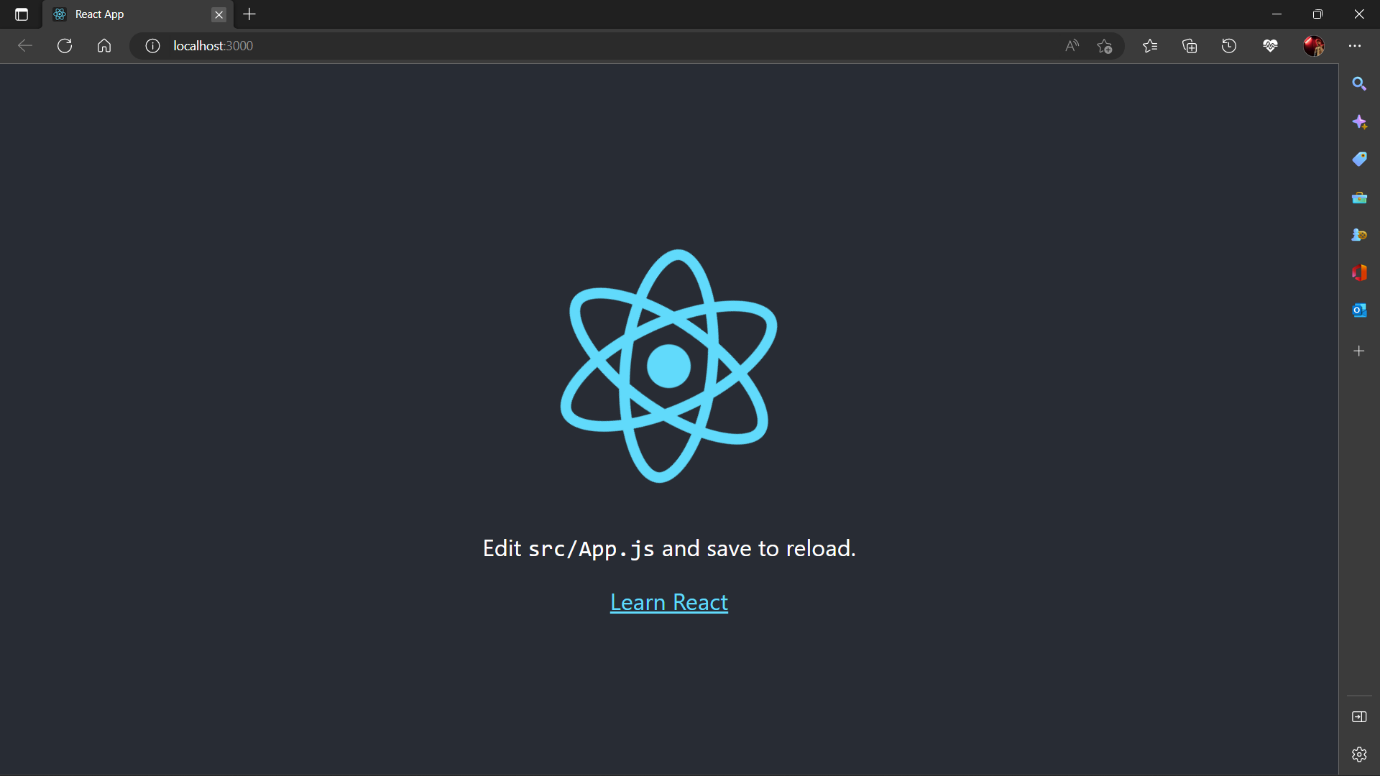
**4.1 REACT APP:**

React App is a command-line interface (CLI) tool that allows developers to quickly and easily create ReactJS applications with minimal configuration. It is based on the Create React App (CRA) project, which was created by the React team at Facebook to streamline the process of creating new ReactJS applications.

React App abstracts away the complexity of setting up a new ReactJS application by providing a pre-configured environment that includes all the necessary tools and dependencies for building and deploying a ReactJS app. It also includes a development server that allows developers to preview their app in the browser without having to set up a separate server.

React App comes with a variety of features, including support for modern JavaScript features such as ES6, ES7, and ES8, as well as support for JSX, a syntax extension for JavaScript that allows developers to write HTML-like code within their JavaScript files. It also includes a wide range of tools and libraries for managing state, routing, and handling data, making it a powerful and flexible tool for building complex web applications.

Overall, React App is a powerful and user-friendly tool that makes it easy for developers to get started with ReactJS and create high-quality, scalable web applications.



***Fig. 4.1 : React App***

**4.2. QR CODE INPUT**

A QR code can contain various types of data, including:

Website URLs

Text messages or notes

Email addresses

Contact information (vCards)

Calendar events

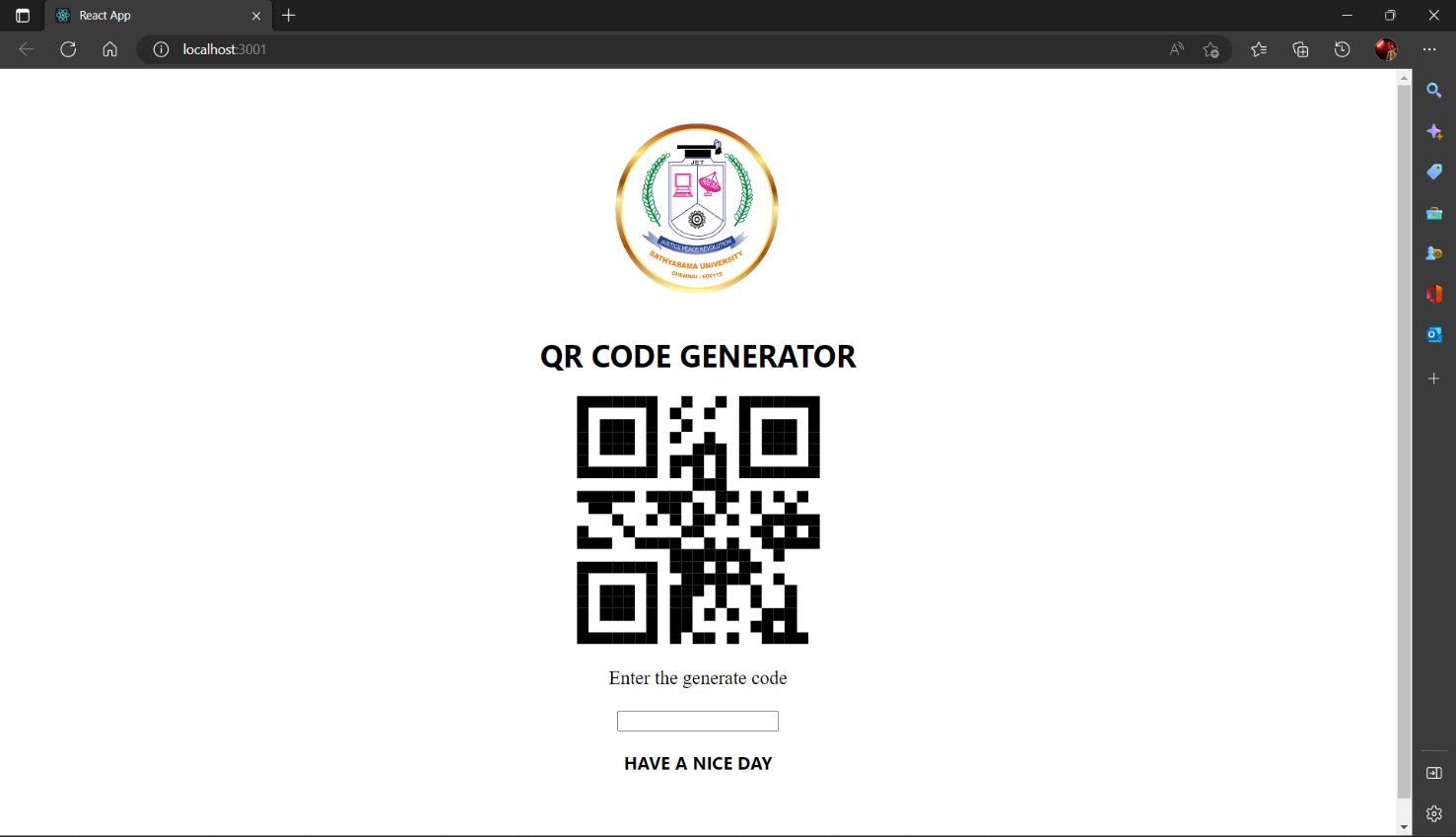
Geographic location coordinates

Wi-Fi network login information

Social media profiles or links

Payment information (e.g., PayPal or Bit coin)

App store links or download information to generate a QR code, you can use a QR code generator tool that allows you to input the desired type of information, and then creates a QR code for you to use.



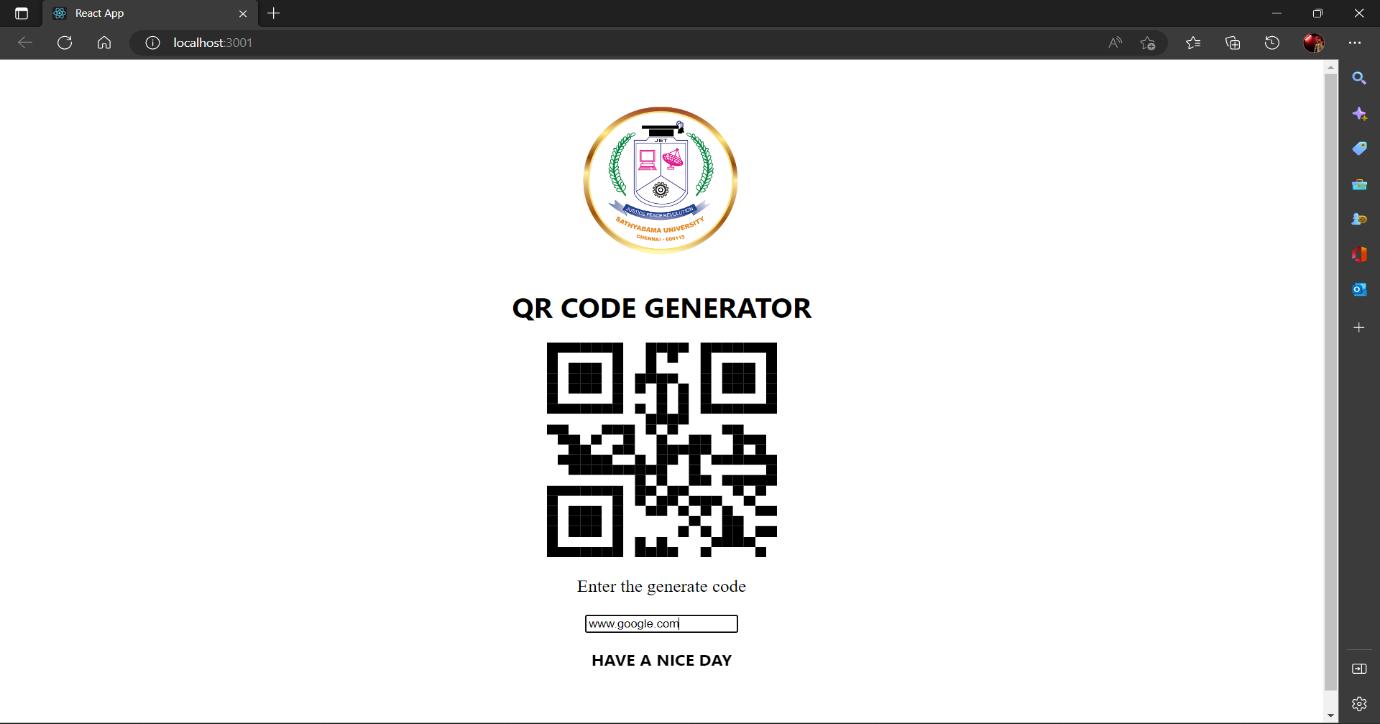
***Fig 4.2 : QR Code Input***

**4.3. QR CODE OUTPUT:**

The output of a QR code is a square or rectangular matrix of black and white modules, which can be scanned using a QR code reader or scanner app on a mobile device or tablet. The scanner app decodes the information contained in the QR code and displays it to the user in a readable format.

For example, if the QR code contains a website URL, scanning the code will open the website on the user's device. If the QR code contains contact information, the scanner app may prompt the user to save the contact information to their phone's address book. Similarly, if the QR code contains a calendar event, the scanner app may add the event to the user's calendar app.

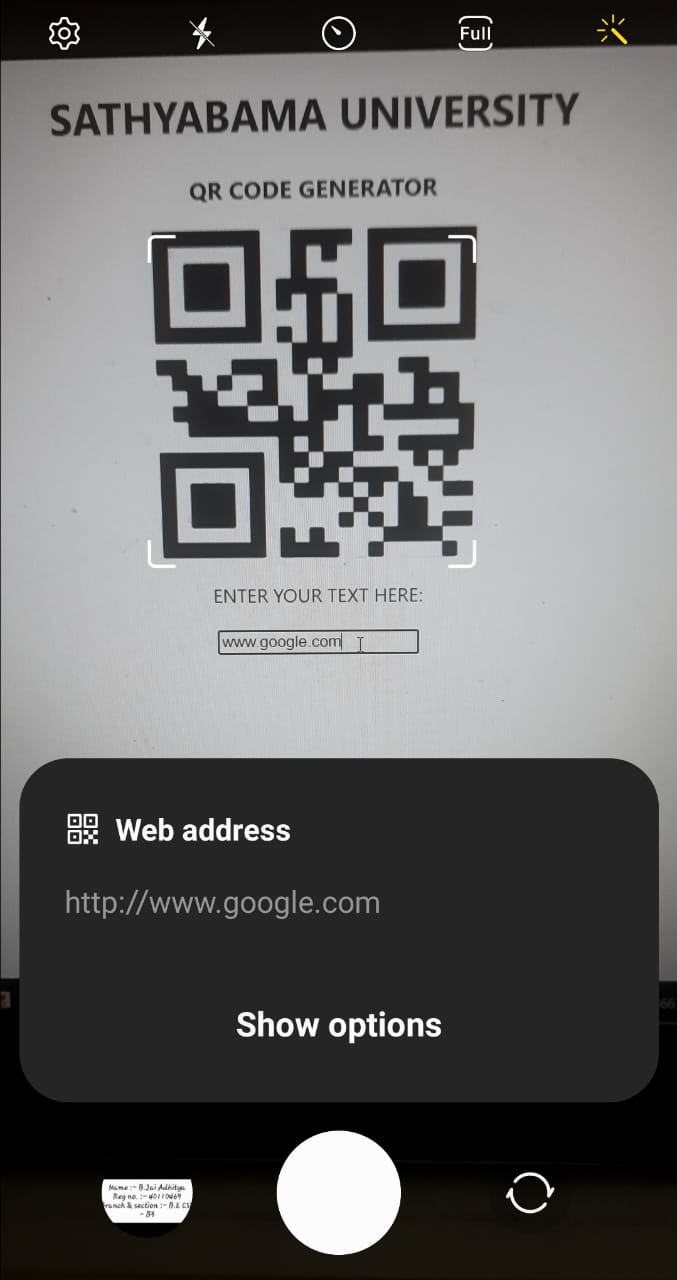
The output of the QR code depends on the type of data encoded within it and the functionality of the scanner app being used to read it.



***Fig 4.3 : QR Code Output***

**4.4. LINK GENERATION:**

The link would be generated after the scanning is done from the smartphone.

***Fig 4.4 : Link Generation***

**CHAPTER 5**

**CONCLUSION**

This application will help the publisher and advertiser to earn money and also the main important agenda is URL shortening which is the very important to browse any websites as they have their own URL address and remembering all the long and numbering URL address is very difficult task to make them easy to remember our application helps the users to remember the URL.

QR code is now being widely used in a variety of businesses. QR code is a way of encoding more information than a traditional bar code. And most importantly, it contains information that can be easily decoded at high speed.

It is shown that how to create the QR codes via the web browser that facilitates users to easily create their own QR codes for websites, emails, business cards, print ads and so on. The proposed method was developed using entirely open source software such as Visual Studio, and python API library.

While developing this project, we learnt:

1. Installing Front end framework like React.js
2. Installing various IDE like VS Code.
3. Using Object oriented language like Javascript.
4. Debugging Javascript bugs.
5. Setting up npm and Node.js.
   1. **FUTURE WORKS**
6. **Customization options:** Allow users to customize the QR code design, including the colors, shapes, and patterns used in the code. This can be achieved by integrating a design tool or by providing users with a set of pre-designed templates to choose from.
7. **Multiple QR code types:** Expand the functionality of the QR code generator to support different types of codes, such as QR codes for Wi-Fi, contact information, or events. This can be achieved by integrating additional libraries or APIs that support these types of codes.
8. **Error correction:** Add error correction to the QR code generator to ensure that codes remain scannable even if they become partially damaged or obscured. This can be achieved by integrating error correction algorithms into the code generation process.
9. **Integration with other applications:** Allow users to easily integrate the QR code generator with other applications and services, such as social media platforms, marketing automation tools, or e-commerce platforms. This can be achieved by providing APIs or integrations with third-party services.
10. **Analytics and tracking:** Add analytics and tracking capabilities to the QR code generator to allow users to track the performance of their codes and gain insights into user behavior. This can be achieved by integrating analytics tools or by building custom analytics features into the application.
11. **Mobile app integration:** Integrate the QR code generator with a mobile app to allow users to easily scan and generate codes on the go. This can be achieved by building a mobile app or by integrating with existing QR code scanner apps.

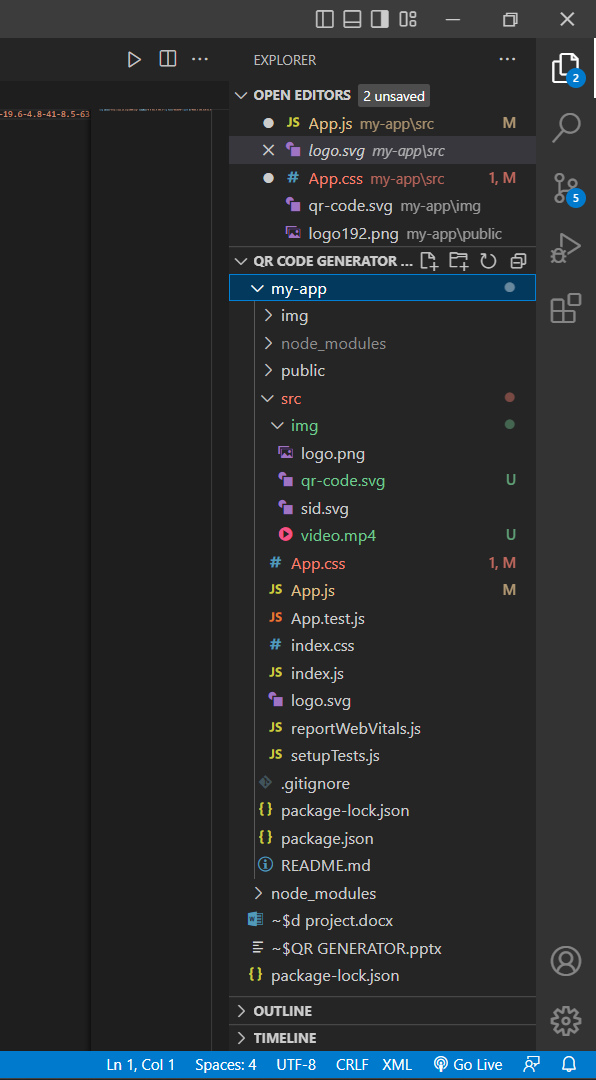
Overall, there are many opportunities for future work on a QR code generator, and the possibilities for expansion and enhancement are almost limitless.

**REFERENCES**

1. Gunter Ollmann; “Security Best Practice: Host Naming & URL Conventions”, Security considerations for web-based applications, Next Generation Security Software Ltd., 2005.
2. Grady Booch, Robert A. Maksimchuk, Michael W. Engle, Bobbi J. Young, Jim Conallen and Kelli A. Houston; “Object Oriented Analysis and Design with Applications” (on AI of cryptanalysis), 3rd edition, Addition- Wesley, 2007.
3. Alexander Neumann ; “Analyzing Security Implications of URL Shortening Services”, Diploma Thesis, RWTH Aachen University - Research Group IT-Security, 2011.
4. Demetris Antoniadis, Iasonas Polakis and Georgios Kontaxis;” we.b: The web of short URLs”, Hyderabad, India March 28 – April 1, 2011.
5. Klien F, Strohmaier M. Short links under attack: geographical analysis of spam in a URL shortener network. Proceedings of the 23rd ACM conference on Hypertext and social media. 2012; 83-88.
6. George N. Mastering Django Core. Birmingham: Packt Publishing. 2016.
7. Reitz K, Schlusser T. The Hitchhiker's Guide to Python: Best Practices for Development. Sebastopol: O'Reilly Media. 2016.
8. Fielding R, Reschke J. RFC 7230 - Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing. June 2014. [Online]. Available: https://tools.ietf.org/html/rfc7230#section-3.1.1. [Accessed 2 May 2018].

**APPENDIX**

**SOURCE CODE**

1. **MY APP DIRECTORY:**
2. **APP.JS:**
3. import logo from './logo.svg';
4. import './App.css';
5. import QRCode from "react-qr-code";
6. import { useState } from 'react';
7. import collegeLogo from './img/logo.png';
8. function App() {
9. const [text, setText] = useState("");
10. function generateQR(e) {
11. setText()
12. }
14. function handleChange(e) {
15. setText(e.target.value)
16. }
17. <div class="max-w-5xl m-auto">
18. <div class="text-xl font-bold text-white">QR Code Generator</div>
19. </div>
20. return (
21. <div classname="App">
22. <div>
23. <img className='logo' src={collegeLogo} alt='college-logo' />
24. </div>
25. <h1>
26. QR CODE GENERATOR
27. </h1>
28. <QRCode value={text} />
29. <div classname="input-here">
30. <p>Enter the generate code</p>
31. <input types="text" value={text} onChange={(e) => { handleChange(e) }} />
32. <form id="generate-form" class="mt-4">
33. <h3 class="my-2">HAVE A NICE DAY</h3>

**3. APP.CSS**

.App {

  text-align: center;

}

body{

  background:0000ff;

}

h1 {

  color: black;

  text-align: center;

}

p {

  font-family: red;

  font-size: 20px;

}

.App-logo {

  height: 40vmin;

  pointer-events: none;

}

@media (prefers-reduced-motion: no-preference) {

  .App-logo {

    animation: App-logo-spin infinite 20s linear;

  }

}

html {

  padding: 3rem 0;

  text-align: center;

}

.primary {

  font-size: 60px;

  color: red;

}

.App-header {

  width: 100%;

  height: 100vh;

  background-color: #dcf308;

  min-height: 100vh;

  display: flex;

  flex-direction: column;

  align-items: center;

  justify-content: center;

  font-size: calc(10px + 2vmin);

  color: #e40d0db9;

}

.App-link {

  color: #0aed71;

}

@keyframes App-logo-spin {

  from {

    transform: rotate(0deg);

  }

  to {

    transform: rotate(360deg);

  }

}

.logo {

  max-height: 200px;

}

**4. INDEX.HTML**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="utf-8" />

    <link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />

    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <meta name="theme-color" content="#000000" />

    <meta

      name="description"

      content="Web site created using create-react-app"

    />

    <link rel="apple-touch-icon" href="%PUBLIC\_URL%/logo192.png" />

    <!--

      manifest.json provides metadata used when your web app is installed on a

      user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/

    -->

    <link rel="manifest" href="%PUBLIC\_URL%/manifest.json" />

    <!--

      Notice the use of %PUBLIC\_URL% in the tags above.

      It will be replaced with the URL of the `public` folder during the build.

      Only files inside the `public` folder can be referenced from the HTML.

      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC\_URL%/favicon.ico" will

      work correctly both with client-side routing and a non-root public URL.

      Learn how to configure a non-root public URL by running `npm run build`.

    -->

    <title>React App</title>

  </head>

  <body>

    <noscript>You need to enable JavaScript to run this app.</noscript>

    <div id="root"></div>

    <!--

      This HTML file is a template.

      If you open it directly in the browser, you will see an empty page.

      You can add webfonts, meta tags, or analytics to this file.

      The build step will place the bundled scripts into the <body> tag.

      To begin the development, run `npm start` or `yarn start`.

      To create a production bundle, use `npm run build` or `yarn build`.

    -->

  </body>

</html>

**5. REPORTWEBVITAL.JS**

const reportWebVitals = onPerfEntry => {

  if (onPerfEntry && onPerfEntry instanceof Function) {

    import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {

      getCLS(onPerfEntry);

      getFID(onPerfEntry);

      getFCP(onPerfEntry);

      getLCP(onPerfEntry);

      getTTFB(onPerfEntry);

    });

  }

};

export default reportWebVitals;

**6. SETUP\_TESTS.JS**

// jest-dom adds custom jest matchers for asserting on DOM nodes.

// allows you to do things like:

// expect(element).toHaveTextContent(/react/i)

// learn more: https://github.com/testing-library/jest-dom

import '@testing-library/jest-dom';