**INTERDISCIPLINARY PROJECT REPORT**

**at**

**Sathyabama Institute Of Science And Technology**

**(Deemed to be University)**

Submitted in partial fulfilment of the requirements for the

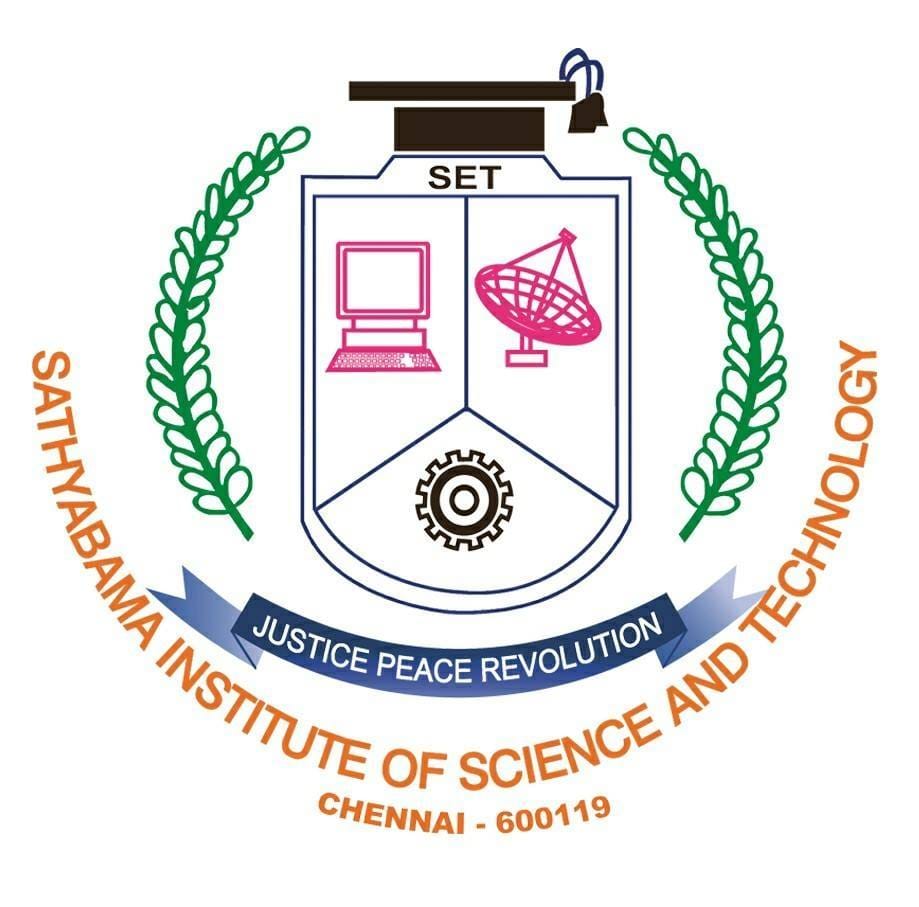
Award of Degree of bachelor of engineering in

Computer science and engineering

By

**ISHANT SINGH**

**(Reg No: 40110465)**



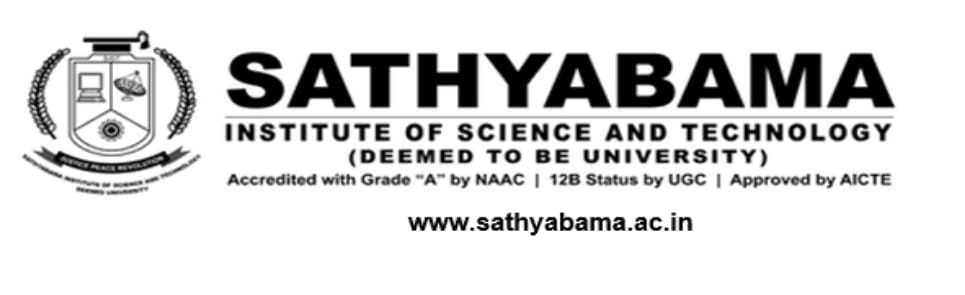
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SCHOOL OF COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI**

**CHENNAI – 600119, TAMILNADU.**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**BONAFIDE CERTIFICATE**

This is to certify that this project Report is the bonafide work of ISHANT SINGH (40110465) who carried out the project entitled “**QR CODE GENERATOR”** under supervision from FEB 2023 to 27 APRIL 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 Singhhereby declare that the Project Report entitled **“QR CODE GENERATOR** “done by under the guidance of Dr. D. Usha Nandini. is submitted in partial fulfillmentof the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

## 

## DATE: 19 - 04 - 2023

## PLACE: Chennai

**SIGNATURE OF THE CANDIDATE**

**ACKNOWLEDGEMENT**

I am pleased to acknowledge my sincere thanks to Board of Managementof SATHYABAMAfor 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.,** Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive review.I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. D. Usha Nandini** for his 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

## C:\Users\Admin\Downloads\Ishant Singh.png

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| CHAPTER NO | TITLE | PAGE NO |
|  | ABSTRACT | 10 |
|  | LIST OF FIGURES | 7 |
|  | LIST OF TABLES | 8 |
| 1. | **INTRODUCTION** | 10 |
|  | 1.1 General | 10 |
|  | 1.2 Literature Survey | 11 |
| 2. | **AIM AND SCOPE OF THE PRESENT INVESTIGATION** | 14 |
|  | 2.1 Objective | 14 |
|  | 2.2 scope | 14 |
|  | 2.3 problem statement | 15 |
| 3. | **EXPERIMENTAL OR MATERIALS AND METHODS ALGORITHMSUSE** | 17 |
|  | 3.1 Existing system | 17 |
|  | 3.2 Proposed system  3.3 Modules  3.4 Architecture diagram | 18 |
| 4. | **RESULTS AND DISCUSSION, ERFORMANCE ANALYSIS** | 19 |
|  | 4.1 Architecture diagram | 19 |
|  | 4.2 Architecture diagram | 20 |
|  | 4.3 input output | 37-38 |
| 5. | **SUMMARY AND CONCLUSION** | 39 |
|  | **REFERENCES**  **APPENDIX** | 40 |
|  | A. SCREENSHOT | 25 - 33 |
|  | B. SOURCE CODE | 9 |

## LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| FIGURE NO | FIGURE NAME | PAGE NO |
| 1.1.0 | QR Code generator | 12 |
| 1.2.0 | Architecture diagram | 19 |
| 2.1.0 | Architecture diagram | 20 |
| 3.1.1 | Input | 36 |
| 3.2.1 | Output | 37 |
| 4.1.0 | Localhost.3000 Input | 12 |
| 4.1.1 | Localhost.3000 Output | 13 |
| 5.0.0 | Open new terminal | 21 |
| 5.0.1 | All right reserved | 24 |
| 5.0.2 | No-Unused-vars | 25 |
| 5.0.3 | On Your Network | 26 |
| 5.0.4 | Install a git repository | 28 |
| 5.1.5 | Install a git repository | 29 |
| 6.0.1 | Npm run eject | 30 |
| 6.0.2 | Package install | 30 |
| 6.0.3 | Npm start | 31 |
| 6.0.4 | Cd react- QR code | 32 |
| 7.0.1 | Webpack complied successfully | 33 |
| 7.0.2 | Localhost.3000 | 34 |
| 7.0.3 | Folder of QR code | 35 |

## LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| TABLE NO | TABLE NAME | PAGE NO |
| 1.1 | Hardware requirement | 21 |
| 1.2 | Software requirement | 21 |
| 2.1 | Folder | 32 |

## 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.

## CHAPTER 1

## INTRODUCTION

## A web address is a string used to identify a web resource. It includes uniform resource locator (URL) and other strings of characters, which can be used to identify web resources when processed appropriately. URL shortener can shorten long URL into shortened one.

## Internet today is full of content and each web page needs its own unique URL. To make a URL identifiable by human, some organization must be applied to make it more structured e.g. by using slashes, date of content creation, slugs and keywords from article and blog posts. However, this makes URL hard to memorize, type manually, or distribute (especially via non-digital media), and can only be copied-pasted for reliability.

## Using the QR codes is one of the most intriguing ways of digitally connecting consumers to the internet via mobile phones since the mobile phones have become a basic necessity thing of everyone.

## In today’s world most part of business required a proper platform to represent their product information for the world for various purposes.

## Some of the reasons are as follows:-

## • Launching new brands

## • Introduction about the product

## • Advertising

## • Marketing

## • Social media campaigns.

## This URL Shortener not only provides a place to short the URL but also helps the client to keep track of the no of clicks on the shortened link. URL Shortener provides the publisher to expand his business through the advertising on the net and shares the website URL very easily and quickly through messages. Publisher is the one who can make use of the URL shortening services and he will be also earning from this services when the users click on his shortened link he will be getting some money that is he will be given money to the total no of clicks on his link.

## 

## *Fig. 4.1.0 Localhost.3000 input*

## 

## *Fig. 4.1.1 Localhost.3000 output*

|  |  |
| --- | --- |
|  |  |
|  | **CHAPTER 2**  **AIM AND SCOPE OF THE PRESENT INVESTIGATION** |

## AIM

## 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.

## SCOPE

## 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.

## 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

## Console as well as from a file of newline separated long URLs.

## 5. The program should work for input files with up to 100000 URLs without crashing.

## CHAPTER 3

## EXPERIMENTAL OR MATERIALS AND METHODS ALGORITHMSUSED

Existing system

URL Shortener is one of the concept which became very famous in the coming year many social websites began to use this services and its one of the very helpful and convenient for the website owners and to the public to access them easily.

In the month of November2009 the URL shortened link of bitly was used by more than two billion times. This URL shortener have helped in the places like when we need to share the website link with clients or with the public it will be very difficult to type the entire URL and it can be sent through the messages because messaging service will be having some restriction on the no of characters so it is good method to use short URL service.

Even we share the URL's through barcode or QR code but we can’t store lengthy URL in a small QR code so the link will be shortened and later it will stored in the QR code for sharing of the link.

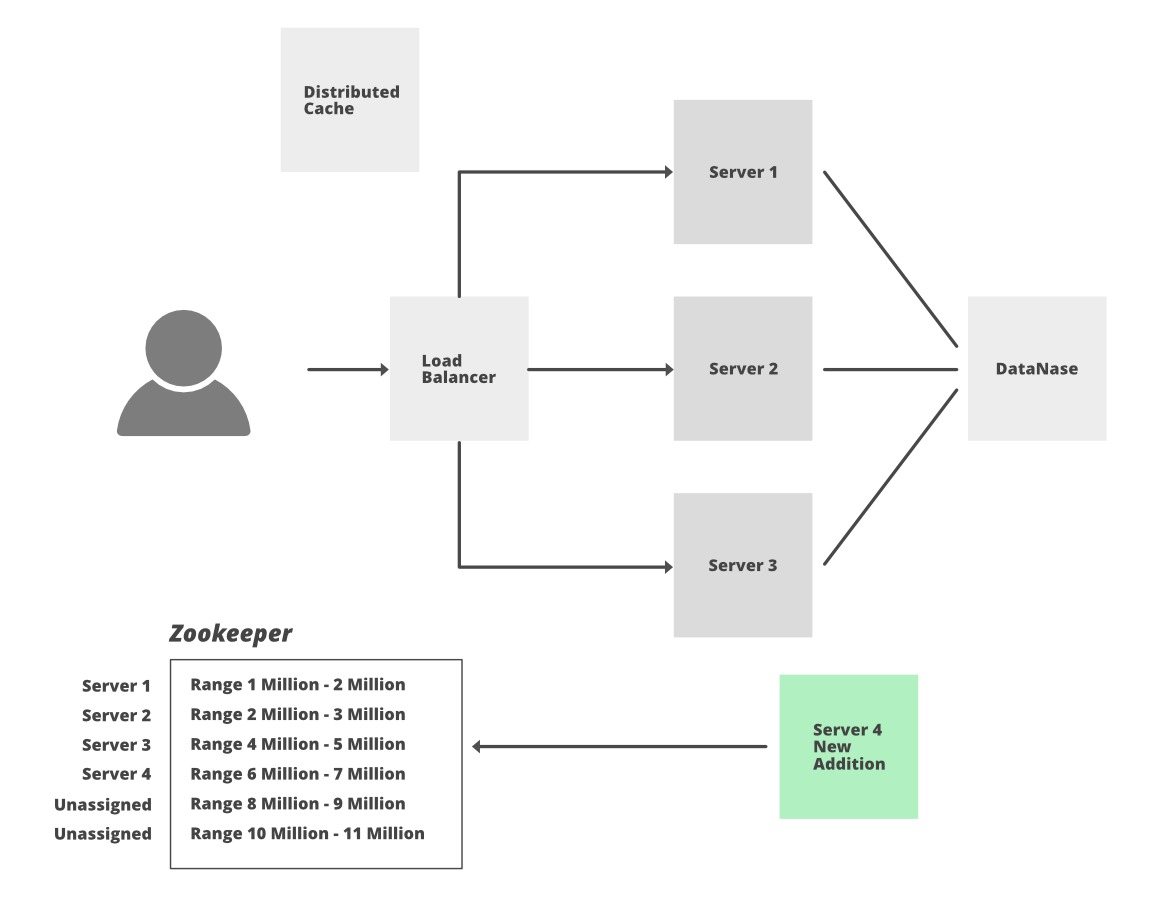
**Proposed system**

There are many numbers of ways in creating short URL. The URL are created by using the base36 which consists of 26 letters and 10 numbers. That is the short URL will be presented as 0 to 9, a to z and it also consists of capital letters and as well as lowercase letter the it will be within base62.

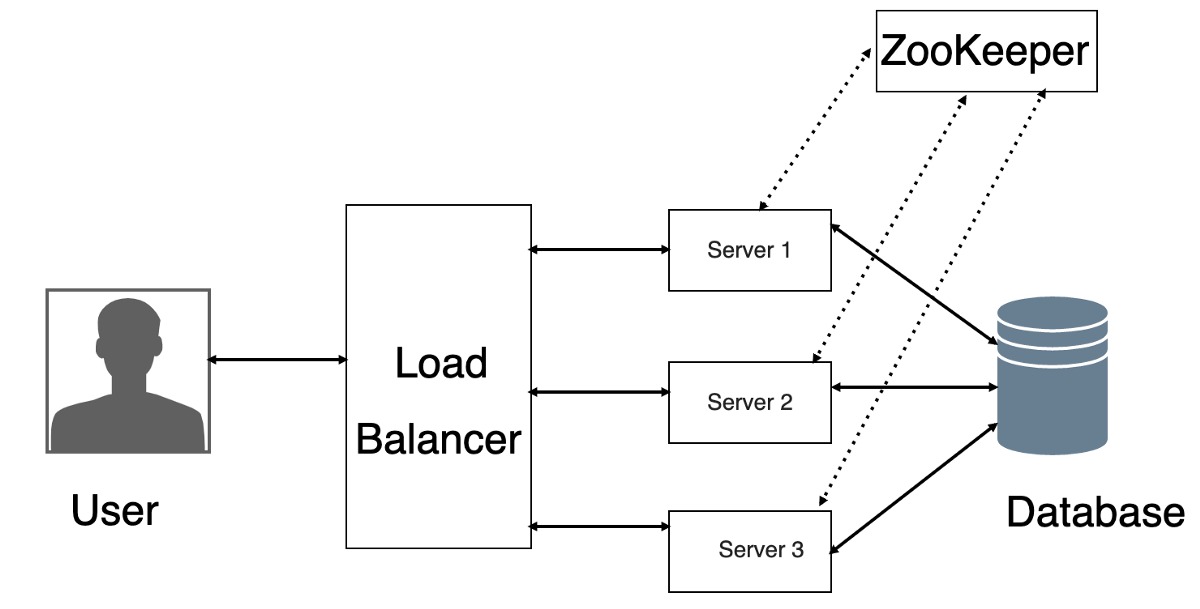
To create the key some numbers generator must be created so that the sequence of key will be unique and it won’t be able to predict or else users will have their own keys to create their short URL. All the protocols are not possible to shorten, some protocols like ftp, http, ftps etc., are being address as URL shortener.

Tiny URL shortening service was the first to start its services and was found and launched in 2002. Google found its own URL shortening service on December 2009 called as goo.gl it was only used for the Google products during that time.

Adf.ly is one of the URL shorteners services which displays ads before going to the really websites the ads are many of popup, banners, interstitial etc.



2.0 Architecture Diagram



2.1 Architecture Diagram

**SYSTEM REQUIREMENTS**

**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

**1.2 Software Requirements:**

Operating System: Windows

VSCode: To handle HTML5, Bootstrap, JavaScript code

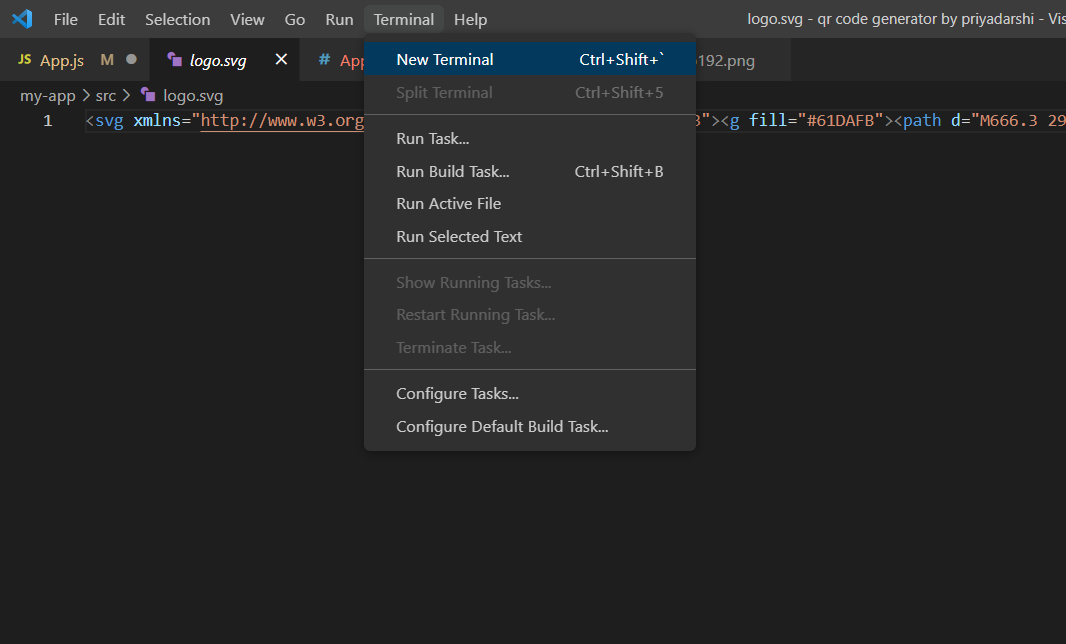
**MODULES**

Names of the Modules:

1. QR Generate Algorithm

2. URL Shortener Base62 Algorithm

First of all you have to create a folder in desktop or any desk, after that you have to open folder in visual studio code then open new terminal in visual studio code.



*Fig. 5.0.0 Open NEW terminal*

## Quick Start

npx create-react-app my-app

cd my-app

npm start

## Creating an App

Npx

npx create-react-app my-app

### npm

### npm init react-app my-app

### Yarn

yarn create react-app my-app

### **Selecting a template**

npx create-react-app my-app --template [template-name]

#### Creating a TypeScript app

**How to Use:**

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

To establish this: I have use Node.js and many other application.

-- to install node/install dependencies: npm install (in console/terminal)

--to install script: npm install react-scripts

--npm audit fix

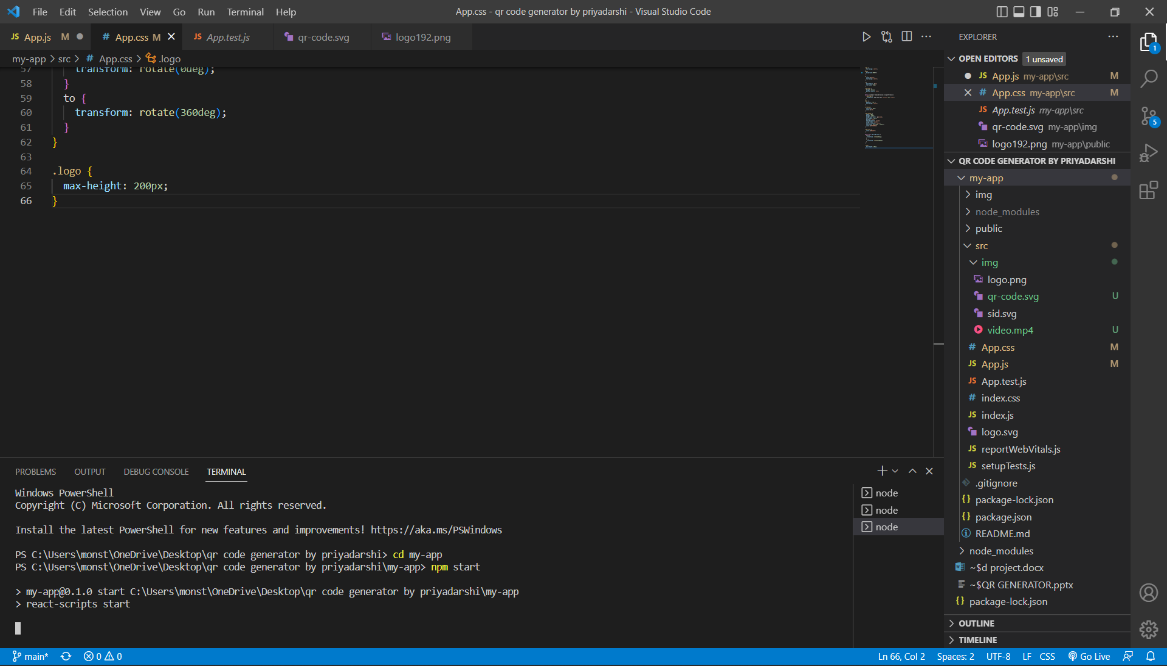
--force (to fix bugs).

--npm install node-fetch

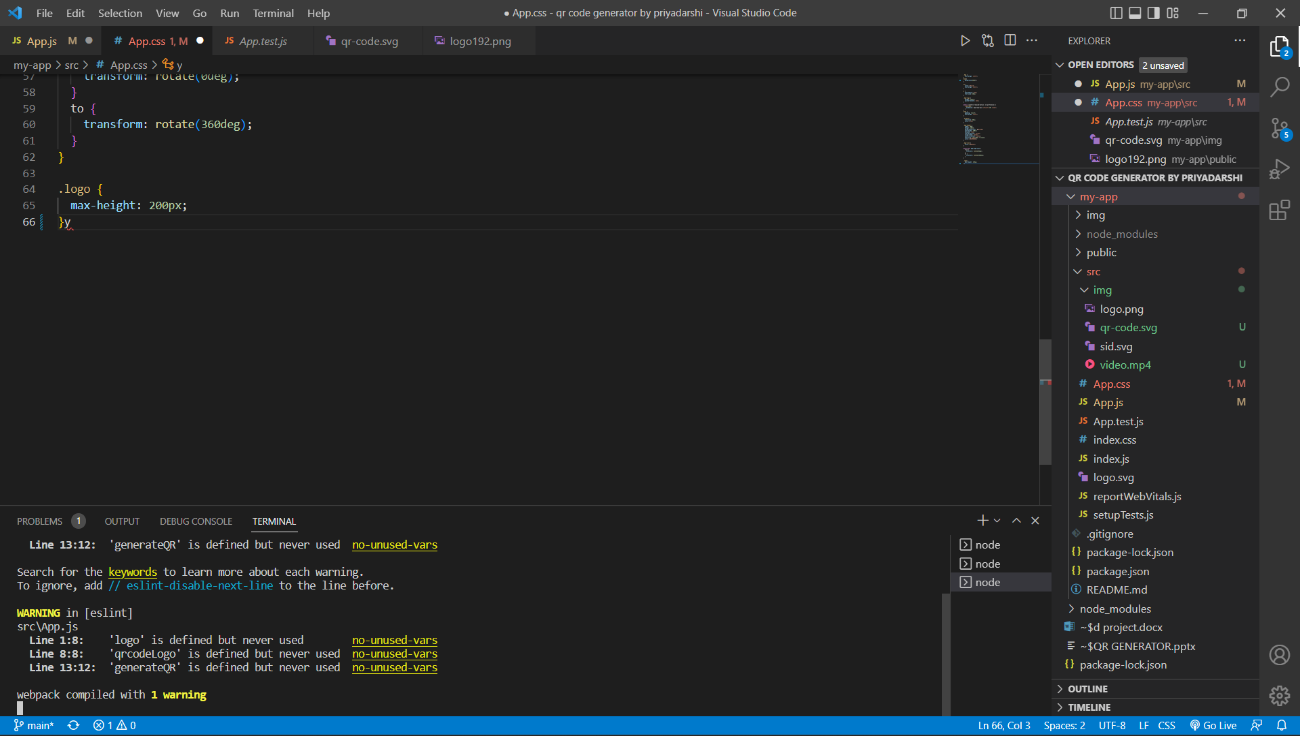
--npm start runs the app in the development mode.

Open Local http://localhost:3000 to view it in the browser.

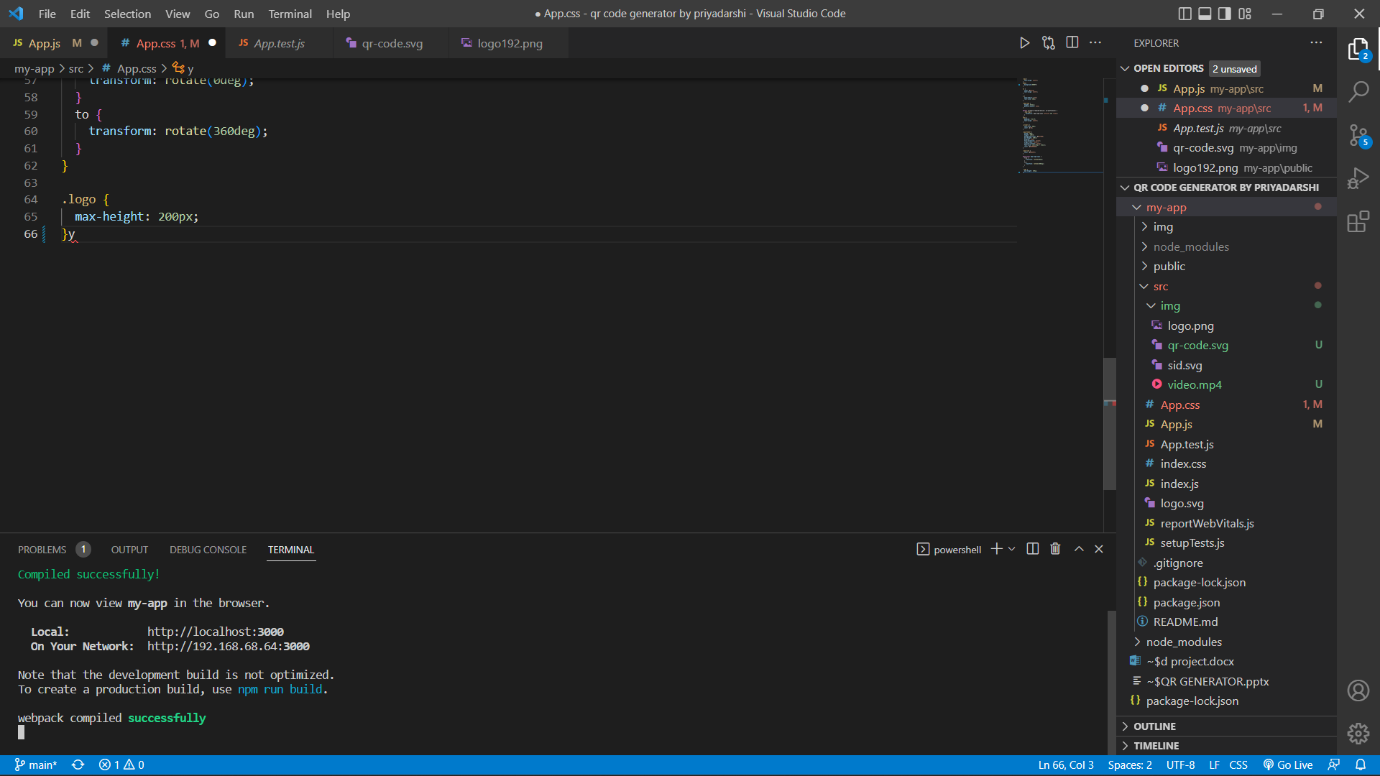
# npx create-react-app my-app --template typescript



*Fig. 5.0.1 All rights reserved*



*Fig. 5.0.2 No-unused-vars*



*Fig. 5.0.3 On your Network*

NOTE: That the development build is not optimized.

To create a production build, use npm run build.

The page will reload if you make edits.

Might see any lint errors in the console.

npm test:

Launches the test runner int interactive watch move

See the section ( <https://create-react-app.dev/docs/getting-started> ) more informations.

npm run build:

Builds the app for production to the build folder.

It correctly bundles React in production mode and optimizes the build for the best performance.

The build is minified and the filenames include the hashes.

# And being ready to deployed. See the section about ( <https://create-react-app.dev/docs/running-tests/> ) for more information.

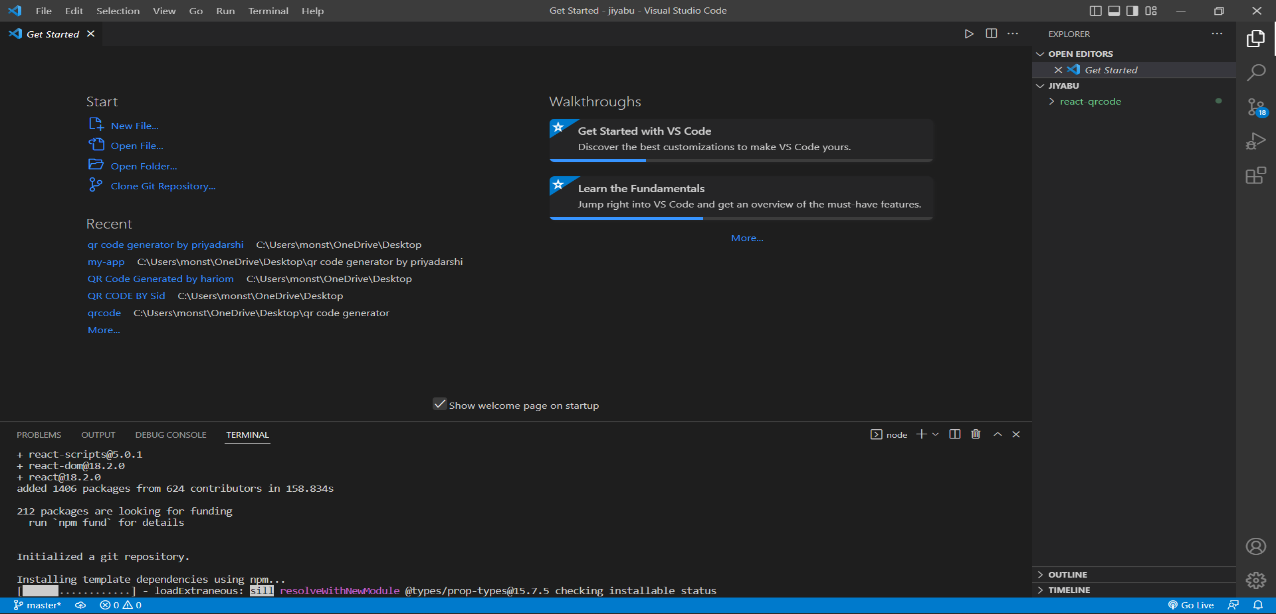
Package install: npm i react-qr-code

Getting Started with react:

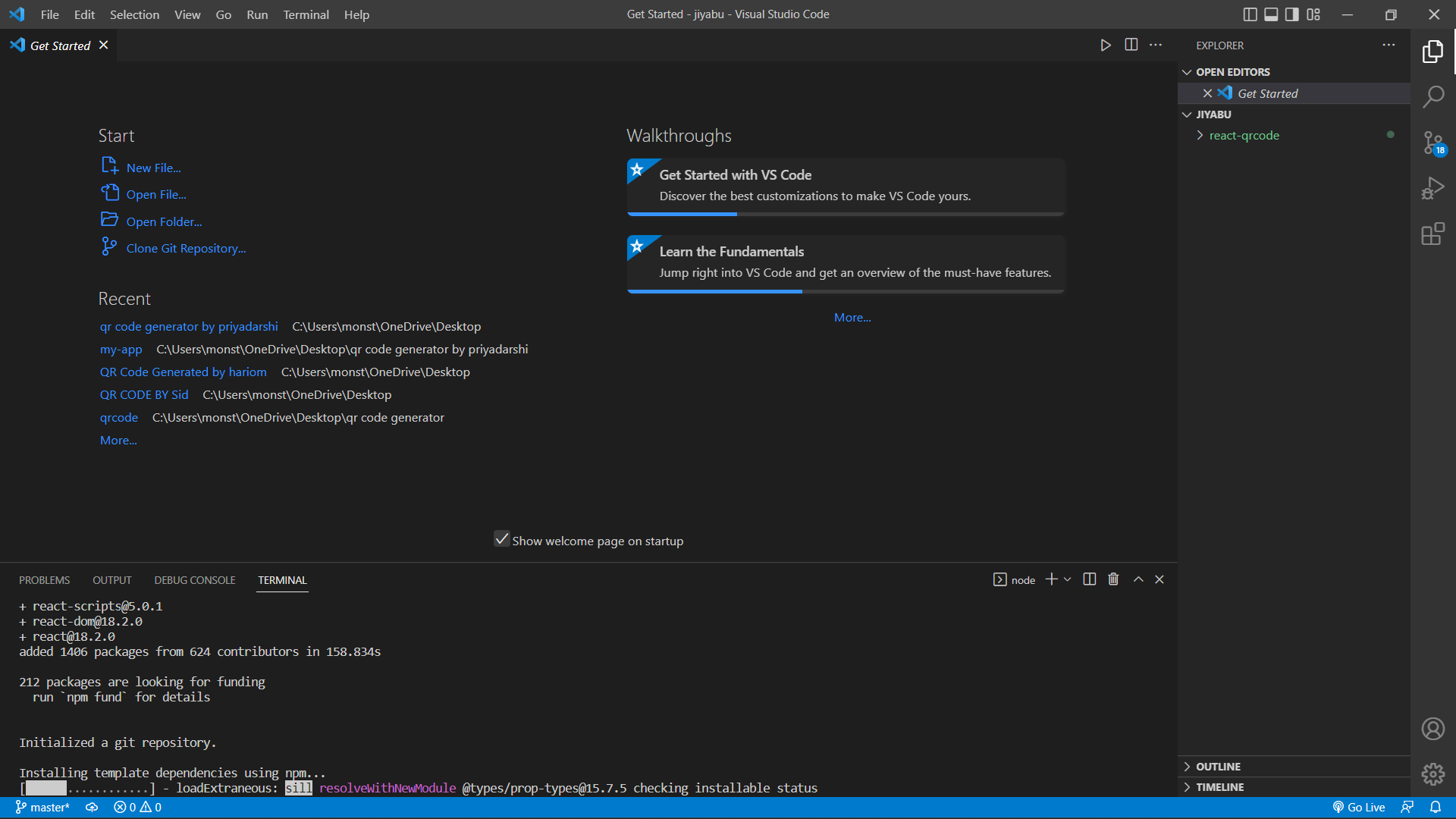
npx create-react-app react- qrcode

cd Qrcode Generator-using-react

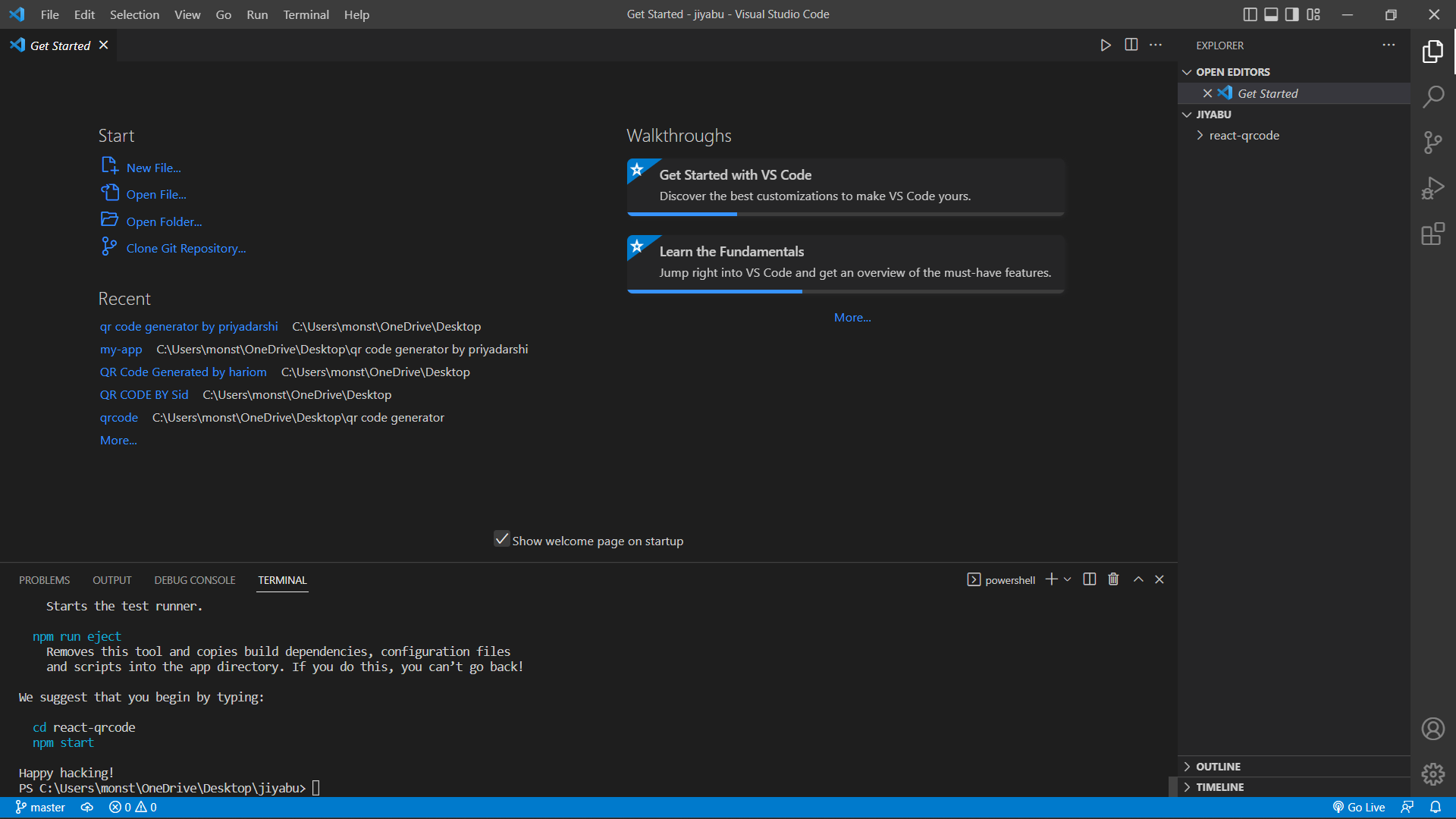
npm start



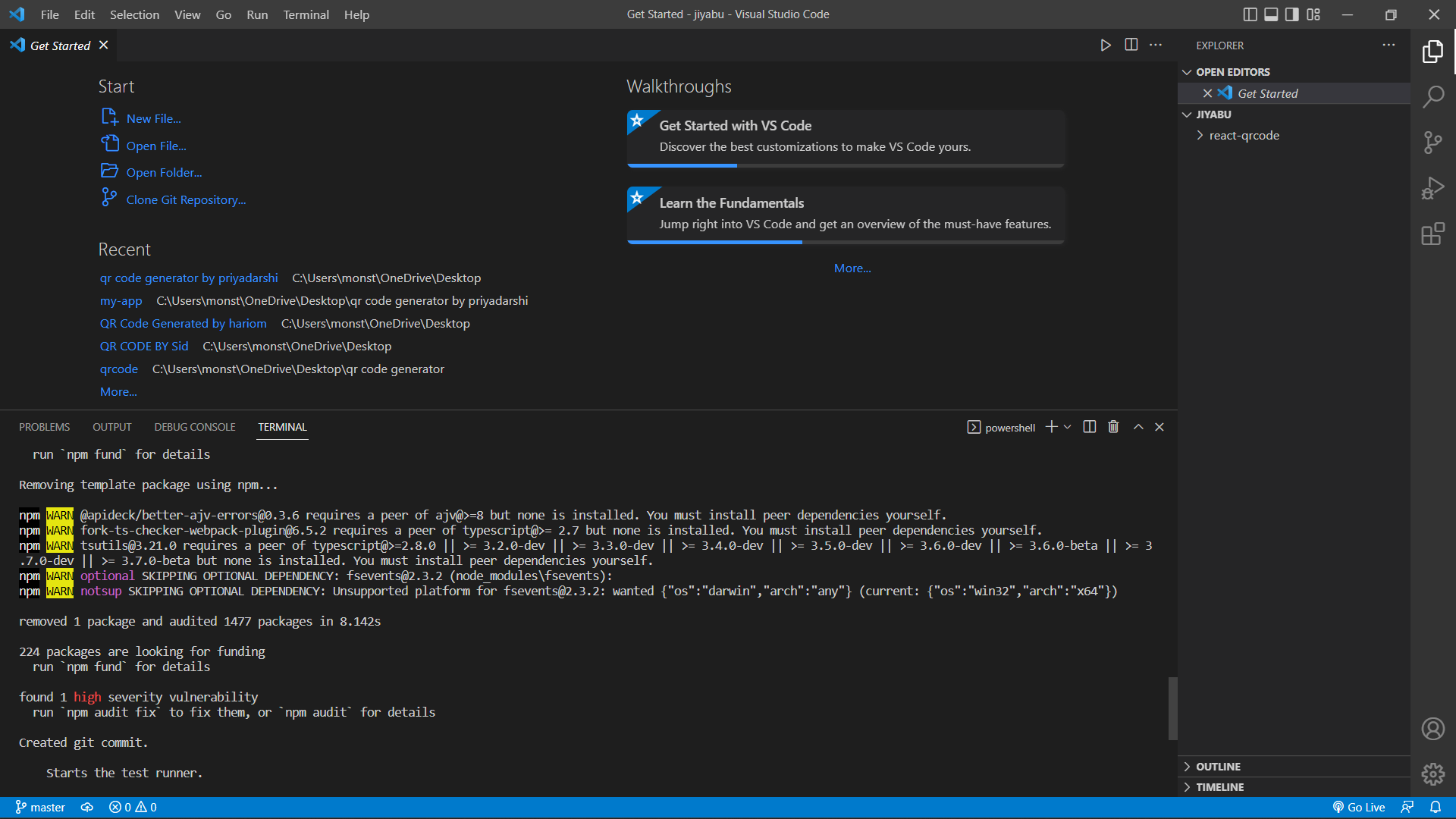
*Fig. 5.0.4 Install a git repository*



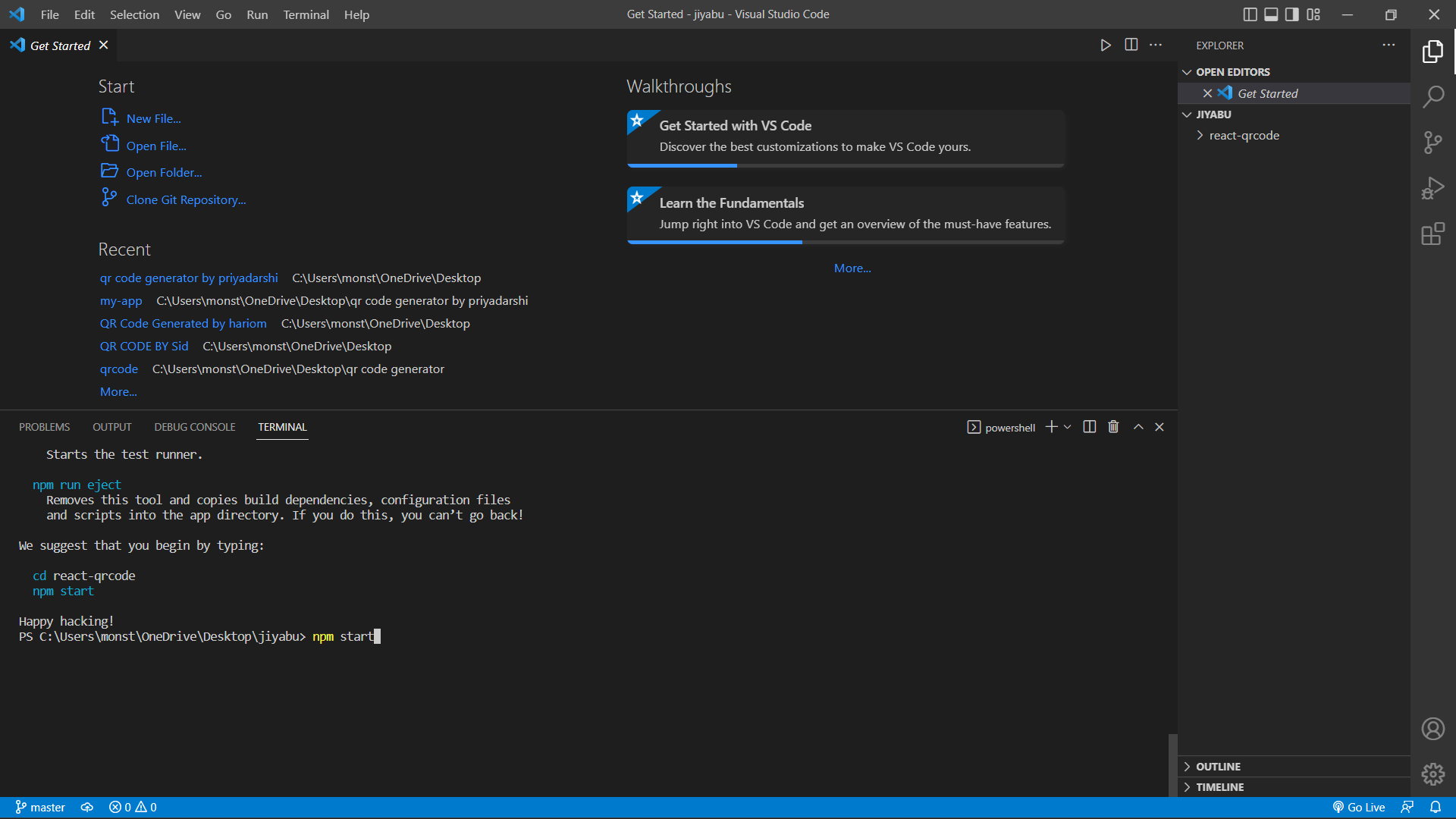
*Fig. 5.1.5 Install a git repository*



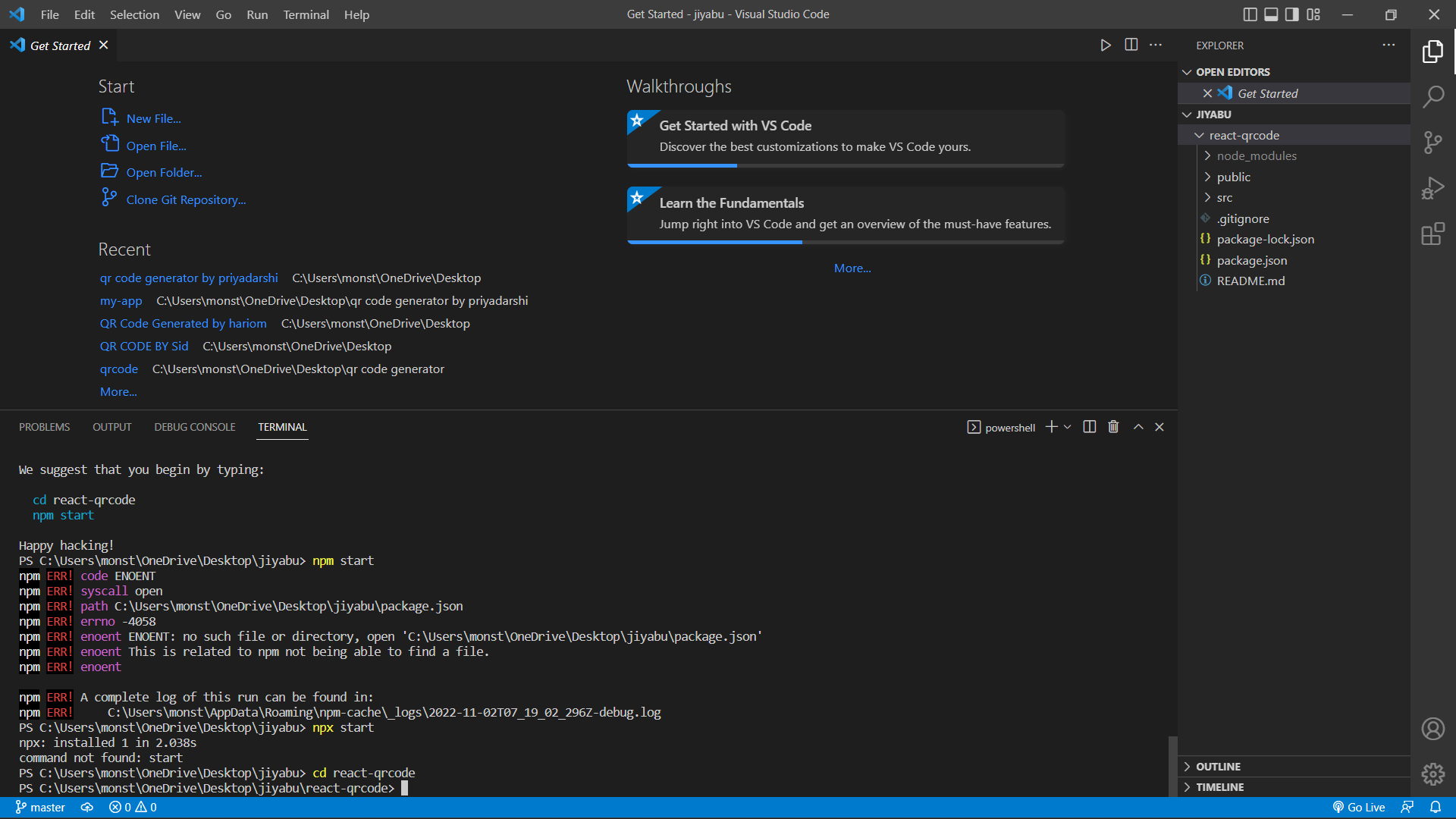
*Fig. 6.0.1 Npm run eject*

**

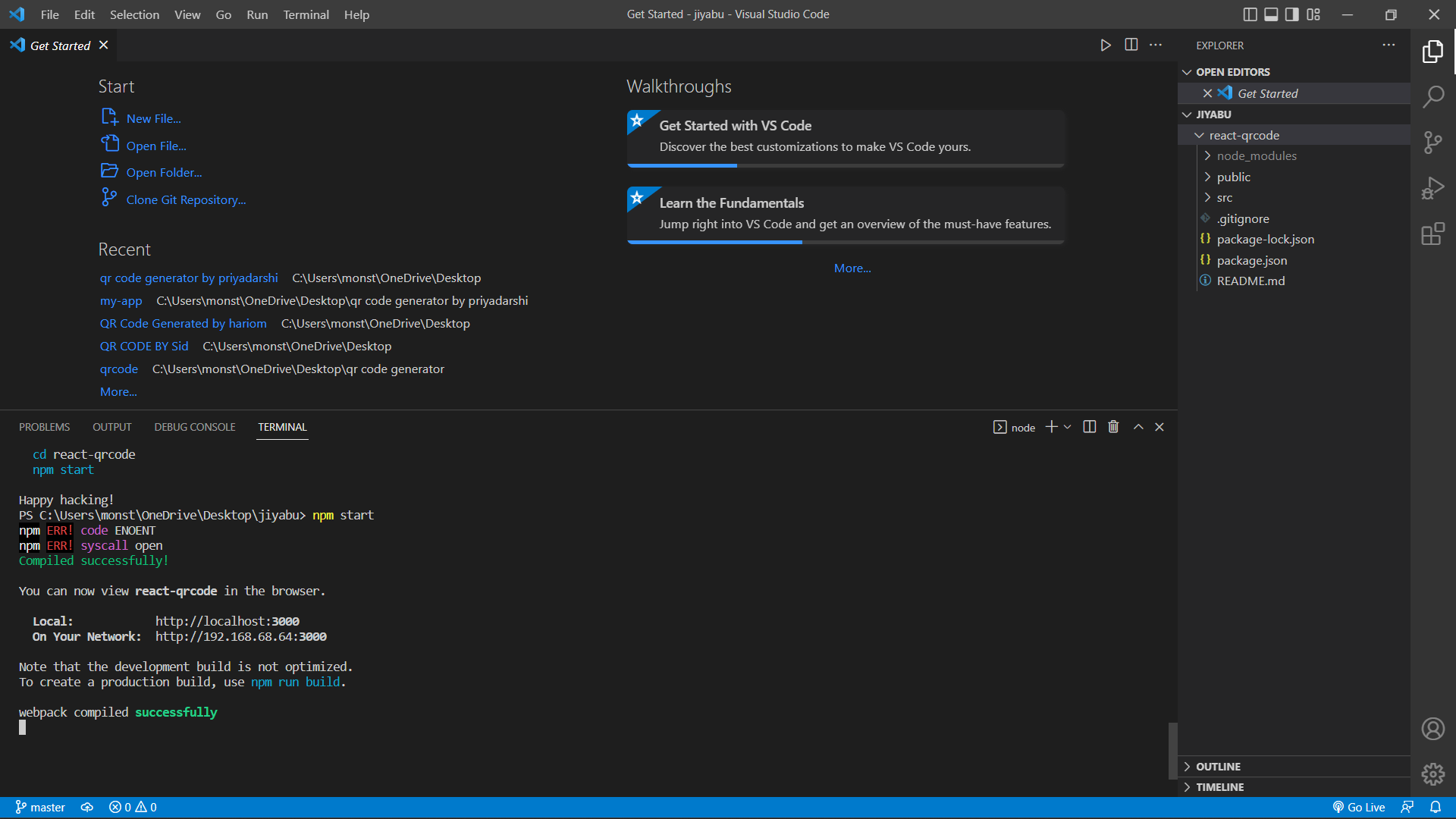
*Fig. 6.0.2 Package install*



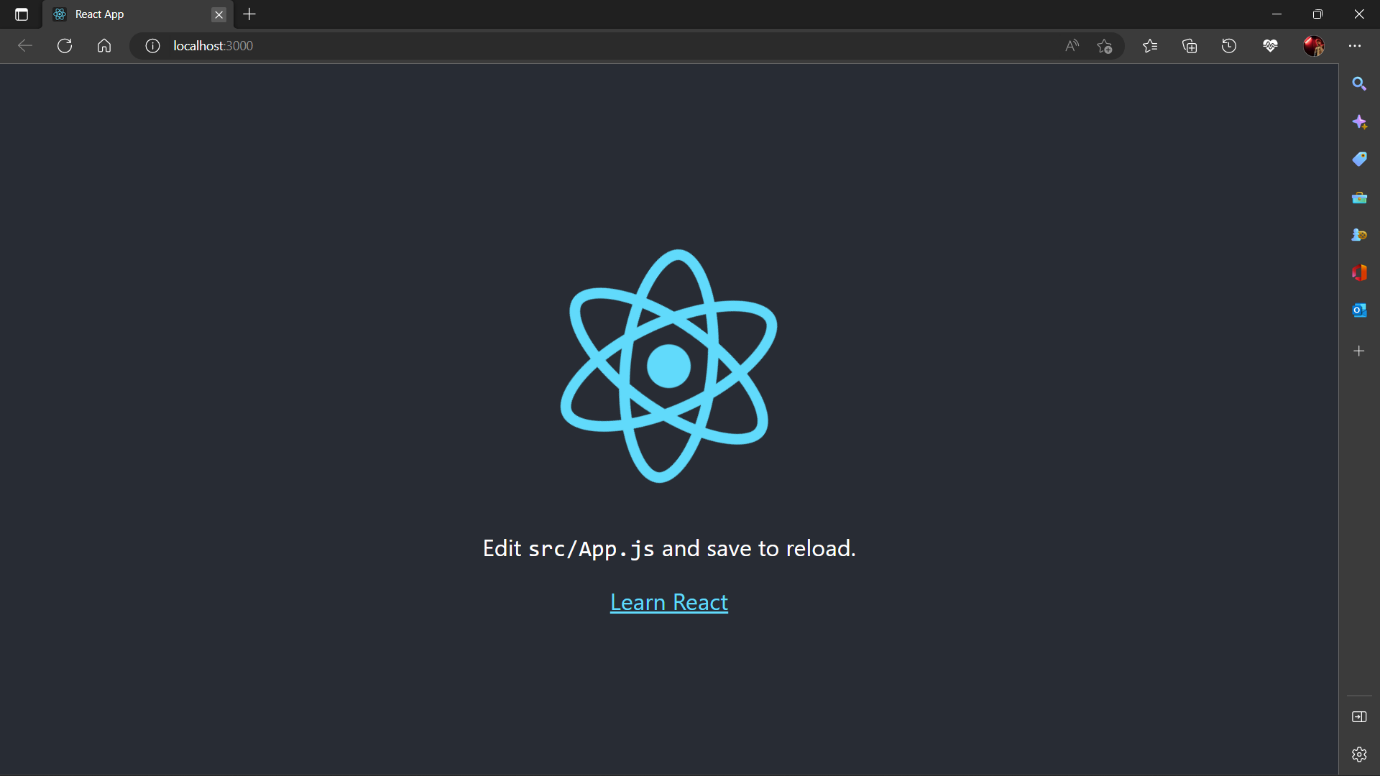
*Fig. 6.0.3 Npm start*



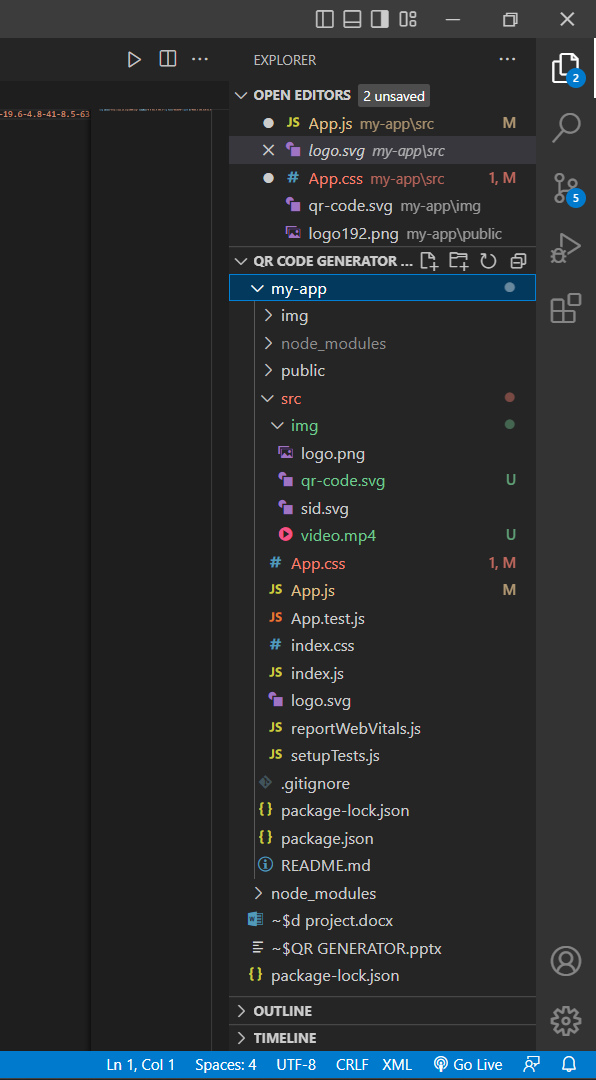
*Fig. 6.0.4 Cd react qr code*



*Fig. 7.0.1 Webpack complied successfully*



*Fig. 7.0.2 Localhost.3000*



**MODULE DESCRIPTION**

1. QRCODE GENERATE

**(App.js)**

import logo from './logo.svg';

import './App.css';

import QRCode from "react-qr-code";

import { useState } from 'react';

import collegeLogo from './img/logo.png';

function App() {

  const [text, setText] = useState("");

  function generateQR(e) {

    setText()

  }

  function handleChange(e) {

    setText(e.target.value)

  }

  <div class="max-w-5xl m-auto">

    <div class="text-xl font-bold text-white">QR Code Generator</div>

  </div>

  return (

  <div classname="App">

      <div>

        <img className='logo' src={collegeLogo} alt='college-logo' />

      </div>

      <h1>

        QR CODE GENERATOR

      </h1>

      <QRCode value={text} />

      <div classname="input-here">

        <p>Enter the generate code</p>

        <input types="text" value={text} onChange={(e) => { handleChange(e) }} />

        <form id="generate-form" class="mt-4">

          <h3 class="my-2">HAVE A NICE DAY</h3>

        </form>

      </div>

      <div class="w-1/3 self-center">

        <img src="img/qr-code.svg" alt="" />

      </div>

    </div>

  );

}

export default App;

1. QRCODE GENERATE

(**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);

  }

}

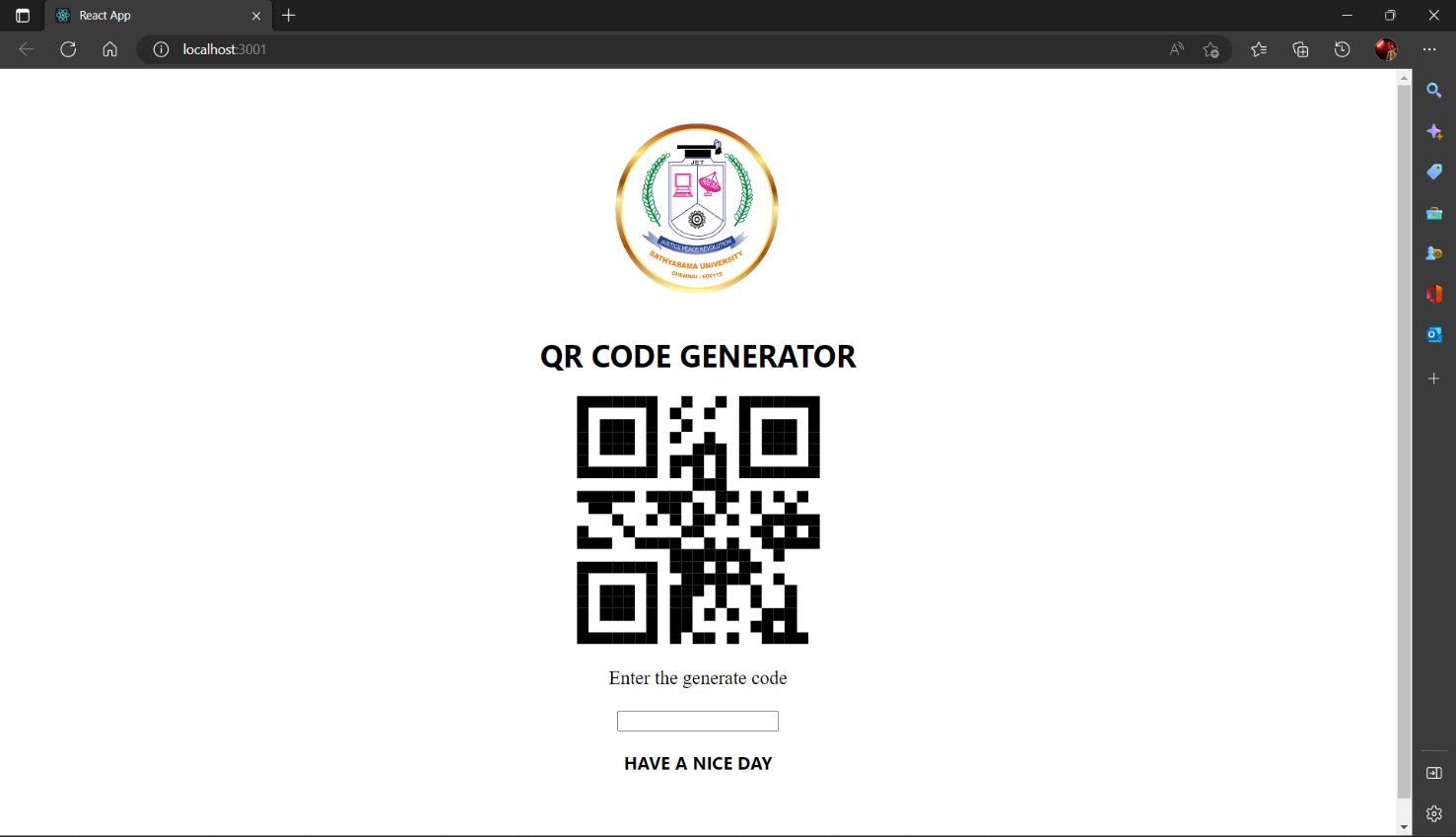
.logo {

  max-height: 200px;

}

**CHAPTER 4**

**RESULTS AND PERFORMANCE ANALYSIS**



*Fig. 3.1.0 input*

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

Website URLs

Text messages or notes

Email addresses

Contact information (vCards)

Calendar events

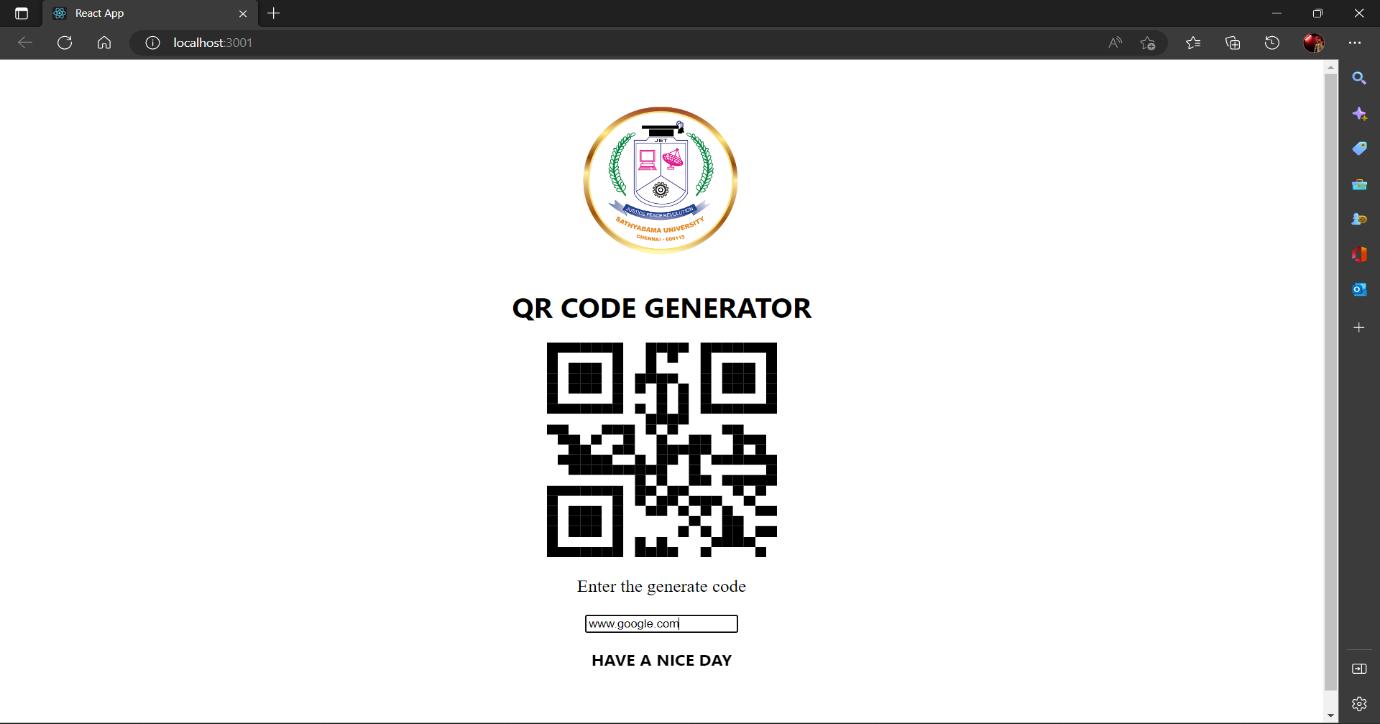
Geographic location coordinates

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.

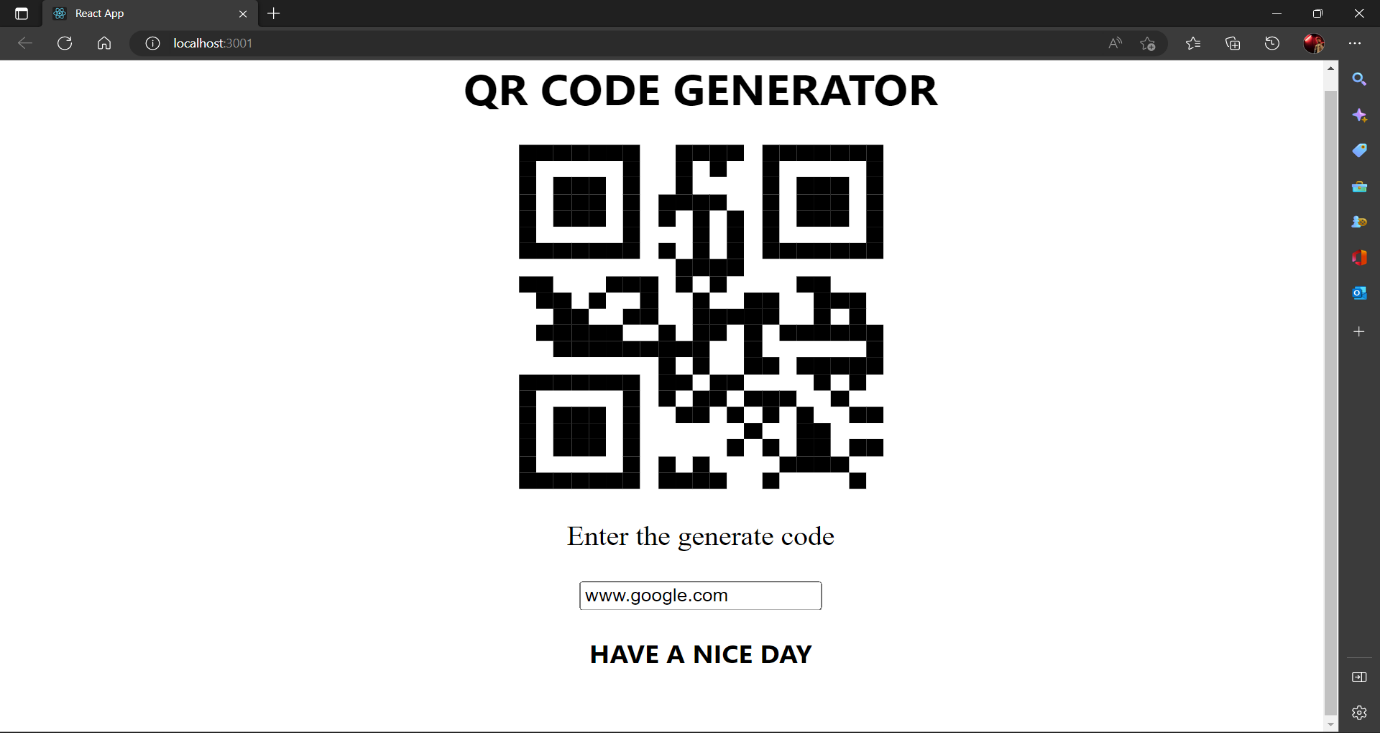


*Fig. 3.2.1 Output*

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.



3.2.2 Code Generate

**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.

**REFERANCES**

Gunter Ollmann; “Security Best Practice: Host Naming & URL Conventions”, Security considerations for web-based applications, Next Generation Security Software Ltd., 2005.

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.

Alexander Neumann ; “Analyzing Security Implications of URL Shortening Services”, Diploma Thesis, RWTH Aachen University - Research Group IT-Security, 2011.

Demetris Antoniadis, Iasonas Polakis and Georgios Kontaxis;” we.b: The web of short URLs”, Hyderabad, India March 28 – April 1, 2011.

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.

George N. Mastering Django Core. Birmingham: Packt Publishing. 2016.

Reitz K, Schlusser T. The Hitchhiker's Guide to Python: Best Practices for Development. Sebastopol: O'Reilly Media. 2016.

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].

**FUTURE WORK:**