

**EasyQR**

**Decode The Future**

Submitted in Partial Fulfillment of Requirements

for the Degree of

**Bachelor of Science (Information Technology)**

By

**Kunal Singh Parmar**

Seat No: 31010921113

Guide

**Prof. Shaikh Bilal Naseem**



**S K Somaiya College**

Somaiya Vidyavihar University

Vidyavihar, Mumbai - 400 077

**2021-24**

**Somaiya Vidyavihar University**



**S K Somaiya College**

**Certificate**

This is to certify that the project report on dissertation entitled EasyQR is bonafide record of the dissertation work done by Kunal Singh Parmar

in the year 2023-24 under the guidance of Prof. Shaikh Bilal Naseem Department of Information Technology and Computer Science in partial fulfillment of requirement for the Bachelor of Science Information Technology degree of Somaiya Vidyavihar University.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Guide / Co-Guide Head of Department/Coordinator

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Director

Date:

Place: Mumbai-77

**S K Somaiya College**

**Certificate of Approval of Examiners**

This is to certify that the project report on dissertation entitled EasyQR is bonafide record of the dissertation work done by Kunal Singh Parmar in partial fulfillment of requirement for the Bachelor of Science Information Technology degree of Somaiya Vidyavihar University.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

External Examiner /Expert Internal Examiner/ Guide

Date:

Place: Mumbai-77

**Somaiya Vidyavihar University**



**S K Somaiya College**

**DECLARATION**

I declare that this written report submission represents the work done based on my and / or others’ ideas with adequately cited and referenced the original source. I also declare that I have adhered to all principles of academic honesty and integrity as I have not misinterpreted or fabricated or falsified any idea/data/fact/source/original work/ matter in my submission.

I understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Signature of the Student**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Name of the Student**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Seat No.**

**Date:**

**Place: Mumbai-77**

**Abstract**



This web application is a useful tool designed to make it easier to connect to a secure WiFi network and easily access certain URLs. The app is designed using JavaScript, HTML, CSS,  React Library and Material UI library to provide a great experience to users looking for a fast and secure connection.

 WiFi network username and password certification allows users to easily generate QR codes  that represent network credentials. Additionally, users can enter any URL they want to generate the corresponding QR code that will transfer the device to the specified website upon scanning. It is designed to simplify the complex process of connecting to a secure WiFi network and accessing the web content you need.

 With its user-oriented approach and effective features, it has become an important tool for  personal and professional use, making every work day easy and productive.

**Contents**



|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No** | **Topic** | | Pg.No |
| **1** | **Introduction** | |  |
|  | 1.1 | Introduction | 7 |
|  | 1.2 | Description | 7 |
|  | 1.3 | Stakeholders | 8 |
| **2** | **Literature Survey** | |  |
|  | 2.1 | Description of Existing System | 10 |
|  | 2.2 | Limitations of present system | 11 |
| **3** | **Methodology** | |  |
|  | 3.1 | Gantt chart (Timeline) | 14 |
|  | 3.2 | Technologies Used and their Description | 15 |
|  | 3.3 | Entity-Relationship Diagram | 17 |
|  | 3.4 | Class Diagram | 18 |
|  | 3.5 | Sequence Diagram | 19 |
|  | 3.6 | Use Case Diagram | 20 |
|  | 3.7 | State Diagram | 21 |
| **4** | **Implementation** | |  |
|  | 4.1 | Software and Hardware Requirements | 22 |
|  | 4.2 | System Coding | 23 |
|  | 4.3 | Screen Layouts and Report Layouts | 37 |
| **5** | **Conclusion and Future Work** | |  |
|  | 5.1 | Conclusion | 39 |
|  | 5.2 | Future Work | 39 |



**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Name of the Figure** | **Page No.** |
| 3.1 | Gantt Chart (Timeline) | 14 |
| 3.3 | ER Diagram | 17 |
| 3.4 | Class Diagram | 18 |
| 3.5 | Sequence Diagram | 19 |
| 3.6 | Use Case Diagram | 20 |
| 3.7 | State Diagram | 21 |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Name of the Table** | **Page No.** |
| 4.1 | Software and Hardware Requirements | 22 |

**Chapter 1**

**Introduction**

*This chapter includes the Introduction , Description and Stakeholders.*

### Introduction of the project

In an increasingly connected world, access to uninterrupted WiFi networks and online content has become an integral part of our daily lives. However, working through the complexities of  network security procedures and URL access is often laborious and time consuming.

To overcome this challenge, we have created various web applications that simplify the process of connecting to a secure WiFi network and easily access specific URLs. Based on  the library and UI component library, our web application provides user interaction and powerfulfunctionality. Whether you're connecting to a WiFi network at a coffee shop or sharing a URL with friends and colleagues, our apps simplify the process with an intuitive and efficient design.

Our app creates a QR code that encodes this information, including security type, username and password. Similarly, users can enter any URL they want and a QR code will be generated which,when scanned, will seamlessly redirect the device to the website. New solutions to connect to  WiFi networks and access the web easily, safely and efficiently.

### Description of the project

Our web application revolutionizes the way users connect to Wi-Fi networks and access specific URLs by offering a seamless and intuitive experience. Using modern web technologies such as JavaScript, HTML, CSS, React libraries and Material UI libraries, our application is designed to simplify complex processes and increase user comfort.

Our core application allows users to enter the necessary credentials for secure Wi-Fi networks, including security type, username and password. With this information, the app generates a QR code that encapsulates the network credentials and allows users to easily share and connect to the network using their mobile devices.

In addition to connecting to a Wi-Fi network, our app also provides a convenient solution for accessing specific URLs. Users can enter any URL of their choice, be it a web page, document or multimedia resource. The app then generates a corresponding QR code that, when scanned, seamlessly redirects the device to the specified URL, eliminating the need for manual typing or navigation.

Simplicity and efficiency make our app suitable for a wide range of use cases, from personal connectivity needs to professional setups. Whether you're sharing Wi-Fi access with guests, connecting to networks in public spaces, or broadcasting URLs at meetings or events, our app simplifies the process and increases productivity.

With its user-centered design and robust functionality, our web application sets a new standard for connecting to Wi-Fi networks and accessing web content. Join us and understand the future of digital connectivity with our innovative solution

**1.3 Stakeholders**

As the sole developer and shareholder of this web application, I have a personal interest in its success and continuous improvement. Here are some key points to consider from my perspective as a stakeholder:

* Vision and goals:

My vision for the web application is to provide users with a seamless and efficient solution to connect to Wi-Fi networks and access specific URLs. I try to streamline these processes to increase user comfort and productivity.

The primary goal is to create a user-friendly app that simplifies complex tasks like entering Wi-Fi credentials and sharing URLs, ultimately improving the overall user experience.

User experience:

User experience is paramount. I prioritize intuitive design and functionality to ensure users can easily navigate the app and complete their tasks with minimal effort.

Continuous improvement based on user feedback is necessary to solve any usability issues and optimize the application for maximum user satisfaction.

* Security and Privacy:

As a developer and stakeholder, I understand the importance of prioritizing security measures to protect user data, especially when handling sensitive information such as Wi-Fi credentials.

Implementing robust encryption protocols and data protection measures is critical to protecting user privacy and maintaining trust in the application.

* Constant improvement:

I am committed to continuous development and improvement of the web application. Regular updates and improvements will address user feedback, fix any bugs or issues, and introduce new features to improve functionality.

Staying abreast of advancements in web technologies and industry trends will ensure the application remains relevant and competitive in an ever-evolving digital environment.

* Marketing and Promotion:

While my primary focus is development, I recognize the importance of promoting a web application to reach a wider audience. Strategies such as social media promotion, targeted advertising, and involvement in online communities will help increase awareness and drive adoption.

* Support and maintenance:

Providing reliable support channels and ongoing maintenance is critical to ensuring a positive user experience. I am committed to solving user queries, solving problems quickly and maintaining the app's functionality over time.

As the only person involved, I was fully invested in the success and growth of the web application. By prioritizing user experience, security, and continuous improvement, I strive to create a valuable tool that meets users' needs and exceeds their expectations.**Chapter 2**

**Literature Survey**

*This chapter consists description of Existing System and limitations of present system.*

**2.1 Description of Existing System**

The existing system represents a robust and user-friendly web application developed using a combination of JavaScript, HTML, CSS, React libraries, and Material UI libraries. This application serves as a convenient tool for facilitating the connection to secure Wi-Fi networks and accessing specific URLs with ease.

* Wi-Fi Network Connectivity:

Users can input essential credentials such as the security type, username, and password of Wi-Fi networks into the application.

The application processes this information to generate a QR code representing the Wi-Fi network credentials.

* URL Generation:

Users have the flexibility to input any type of URL into the application, including websites, documents, or multimedia content.

Upon submission, the application generates a QR code corresponding to the provided URL.

* QR Code Scanning:

Users can scan the generated QR codes using their mobile devices equipped with QR code scanning capabilities.

For Wi-Fi network QR codes, scanning facilitates automatic connection to the specified network without the need for manual configuration.

For URL QR codes, scanning redirects the user's device to the corresponding website or web content seamlessly.

* User Interface and Experience:

The application boasts an intuitive user interface designed to streamline the process of inputting credentials and generating QR codes.

Material UI libraries are leveraged to enhance the visual appeal and usability of the application, ensuring a pleasant user experience.

* Compatibility and Accessibility:

The web application is designed to be compatible across various devices and platforms, ensuring accessibility for users with diverse technological preferences.

It is optimized to function seamlessly on both desktop and mobile devices, catering to users' needs regardless of their preferred device.

**2.2 Limitations of Present System**

* Dependency on QR Code Scanning:

The functionality of the web application heavily relies on users' ability to scan QR codes using their devices. Users who are unable to scan QR codes, whether due to technical limitations or lack of familiarity, may face difficulties in utilizing the application.

* Mobile Device Requirement:

Utilizing the QR code scanning feature requires users to have access to a mobile device equipped with a camera. Users without compatible mobile devices may not be able to fully utilize the application's features.

* Compatibility with Wi-Fi Network Security Types:

While the application supports various security types for Wi-Fi networks, there may be compatibility issues with certain network configurations. Users may encounter difficulties connecting to networks with non-standard security settings or protocols not supported by the application.

* Length Restrictions for URLs:

The generation of QR codes for URLs may encounter limitations with excessively long URLs. Users attempting to generate QR codes for lengthy URLs may experience issues with readability or compatibility across different scanning devices.

Accessibility Considerations:

The web application's user interface and functionality may not be fully optimized for accessibility standards, potentially excluding users with disabilities or impairments from accessing and utilizing the application effectively.

* Offline Usage Restrictions:

The application's functionality may be limited or unavailable when users are offline or have restricted internet access. This could pose challenges for users attempting to generate QR codes or access network credentials in environments with limited connectivity.

* Data Privacy Concerns:

Users may have concerns about the privacy and security of the data input into the application, particularly regarding the storage and handling of sensitive information such as Wi-Fi network credentials. Addressing these concerns and implementing robust data protection measures is essential to maintaining user trust.

* Limited User Management Features:

The application may lack advanced features for managing user accounts, permissions, and access control. Organizations or users requiring more sophisticated user management capabilities may find the application's functionality insufficient for their needs.

* Platform Compatibility:

While efforts are made to ensure compatibility across different web browsers and devices, users may encounter inconsistencies or usability issues when accessing the application on certain platforms or devices. Continuously testing and optimizing for platform compatibility is essential to improving user experience.

* Scalability Challenges:

As user demand grows, scalability challenges may arise in handling increased traffic and processing QR code generation requests efficiently. Proactively addressing scalability concerns and optimizing server infrastructure can mitigate potential performance issues.

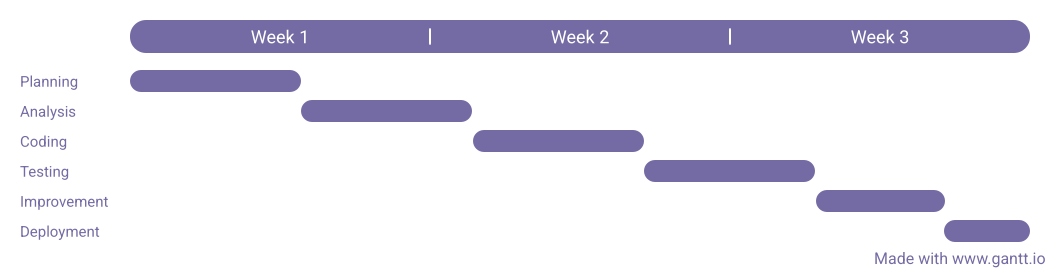
Addressing these limitations is crucial to enhancing the user experience and overall effectiveness of the web application. Continuously seeking user feedback, conducting thorough testing, and implementing iterative improvements are key strategies for mitigating these limitations over time.

**Chapter 3**

**Methodology**

*This chapter shows the Systematic Diagrams and there uses, also it talks about the technologies which are used to make this website Responsive. Each diagram represents the different features of website. Technologies which Play Important role and interactions among it key pages.*

## 3.1 Gantt Chart (Timeline)



*Figure 1. Gantt Chart*

A bar chart that illustrates a project schedule, showing tasks or activities along the horizontal axis and time intervals along the vertical axis. Gantt charts visually display the start and end dates of each task, dependencies between tasks, and the overall project timeline. The chart provides a visual representation of the sequential progression of tasks, allowing for effective project management and monitoring of progress.

**3. 2 Technologies Used and their Description**

* **HTML:**

HTML, or HyperText Markup Language, serves as the foundation of web pages, providing structure and meaning to the content displayed in browsers. It utilizes tags to define elements such as headings, paragraphs, images, links, and more. Through its simplicity and versatility, HTML enables developers to create rich, interactive, and accessible web experiences for users worldwide. As a cornerstone of the web, HTML empowers creators to craft engaging digital content and facilitate seamless communication and interaction across the internet.

* **CSS:**

CSS, or Cascading Style Sheets, is a fundamental component of web development, providing the means to enhance the visual presentation and layout of HTML elements. Through CSS, developers can apply styles such as colors, fonts, spacing, and positioning to create visually appealing and responsive web pages. By separating design concerns from content structure, CSS promotes modularity, maintainability, and consistency in web development projects. With its powerful features and capabilities, CSS empowers designers and developers to craft immersive and engaging user experiences across a wide range of devices and screen sizes, making it an indispensable tool in modern web design.

* **JavaScript:**

JavaScript is a versatile programming language primarily used for adding interactivity and dynamic behavior to web pages. As a client-side scripting language, it runs directly in users' web browsers, enabling developers to create responsive and interactive web applications. JavaScript empowers developers to manipulate HTML and CSS, handle user interactions, dynamically update content, and communicate with web servers asynchronously. Its widespread adoption and extensive ecosystem of libraries and frameworks make JavaScript an essential tool for building modern web applications and delivering rich user experiences across various platforms and devices.

* **NodeJS:**

Node.js is a powerful runtime environment built on Chrome's V8 JavaScript engine, enabling developers to run JavaScript code outside the browser. As a server-side platform, Node.js facilitates the development of scalable, high-performance web applications and APIs. Leveraging an event-driven, non-blocking I/O model, Node.js excels at handling concurrent connections and asynchronous operations, making it ideal for real-time applications and data-intensive tasks. Its extensive ecosystem of modules and packages, managed by npm (Node Package Manager), provides developers with a wealth of tools and libraries for building robust and efficient server-side applications. With its speed, scalability, and versatility, Node.js has become a popular choice for modern web development, empowering developers to build fast and scalable applications with JavaScript on both the client and server sides.

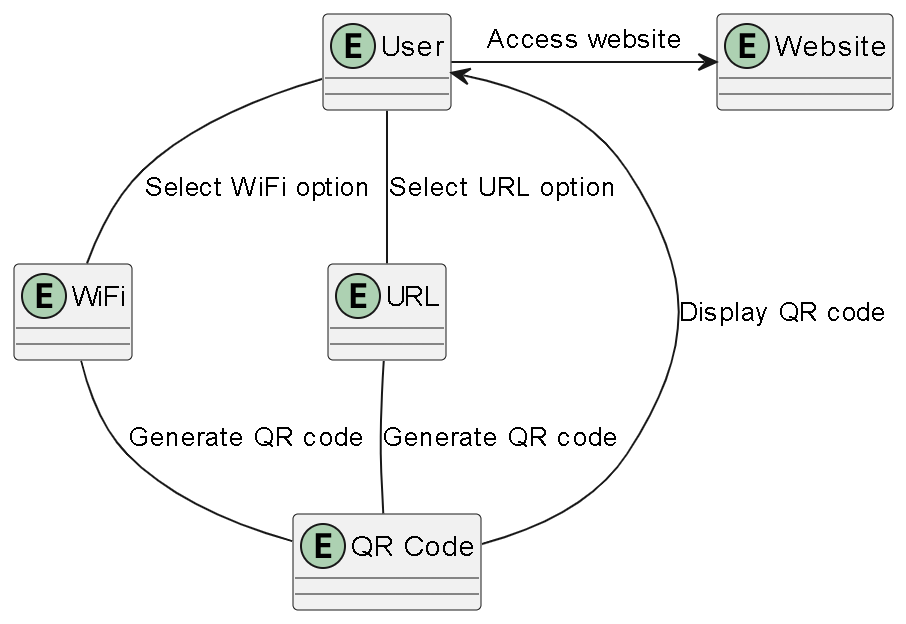
* **Material-UI:**

Material-UI is a popular React UI framework that provides a collection of pre-designed components inspired by Google's Material Design principles. These components offer developers a consistent and visually appealing user interface for their web applications. Material-UI's modular and customizable nature allows developers to easily integrate and style components to match their application's design requirements. With a wide range of components including buttons, cards, inputs, navigation bars, and more, Material-UI simplifies the development process, enabling developers to focus on building functionality rather than designing UI elements from scratch. Its responsive design and accessibility features ensure that applications built with Material-UI are user-friendly and accessible across various devices and platforms. Overall, Material-UI libraries empower developers to create modern, sleek, and intuitive user interfaces for their React-based web applications.

* **React Libraries:**

React's extensive library ecosystem offers developers a wide range of tools and utilities to streamline development processes and enhance application functionality. From state management to routing, data fetching, and UI components, these libraries provide valuable resources for building modern and efficient React applications.

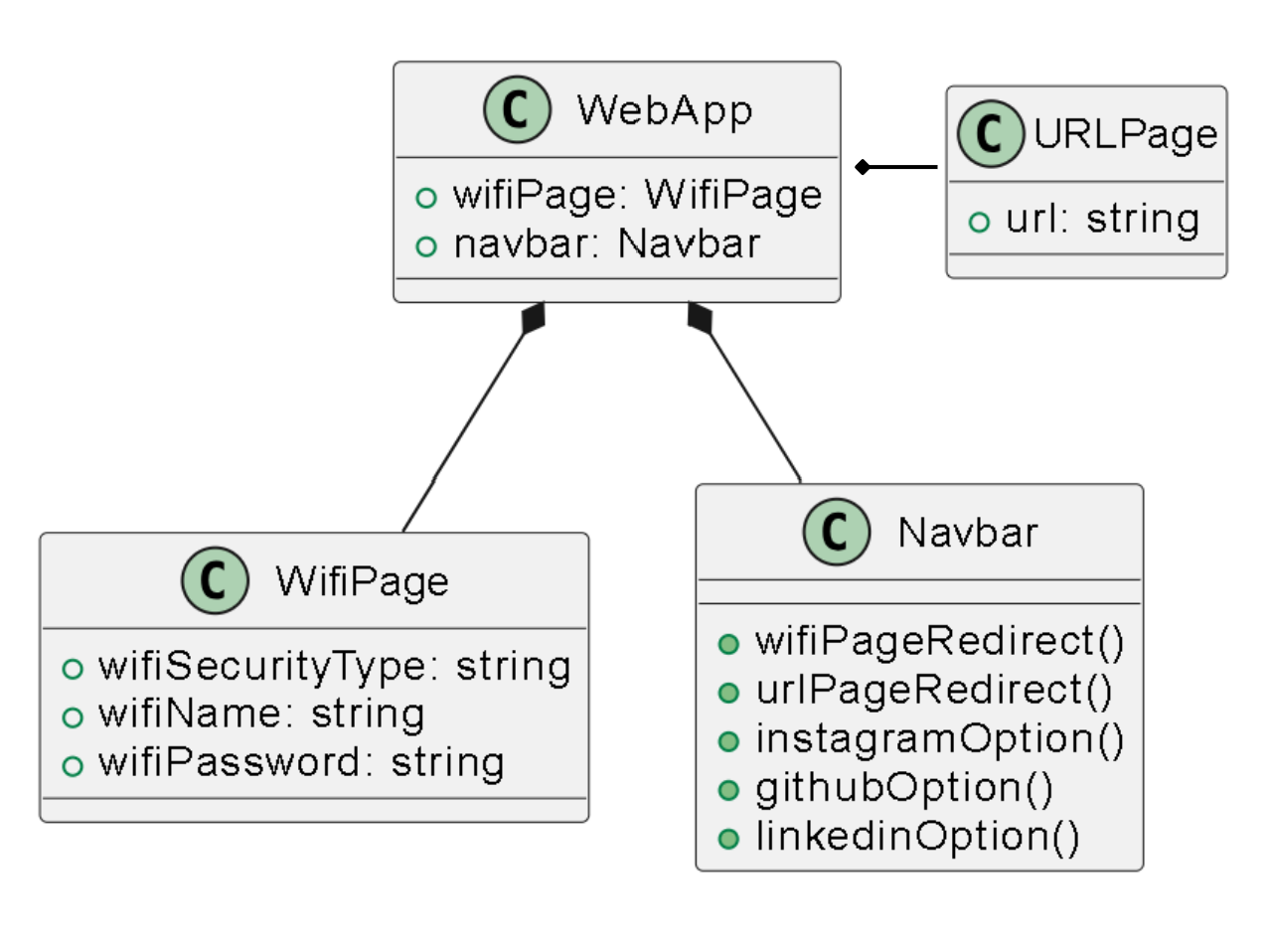
**3.3 Entity Relationship Diagram**

****

*Figure 2. ER Diagram*

An Entity-Relationship (ER) diagram is a visual representation of the relationships between entities in a database schema. It illustrates the structure of the database by depicting entities as boxes and relationships between them as lines. Entities represent objects or concepts, while relationships denote how entities are connected or associated with each other. ER diagrams are invaluable tools for database designers and developers, as they provide a clear and concise overview of the database schema, helping to ensure proper data organization, integrity, and consistency.

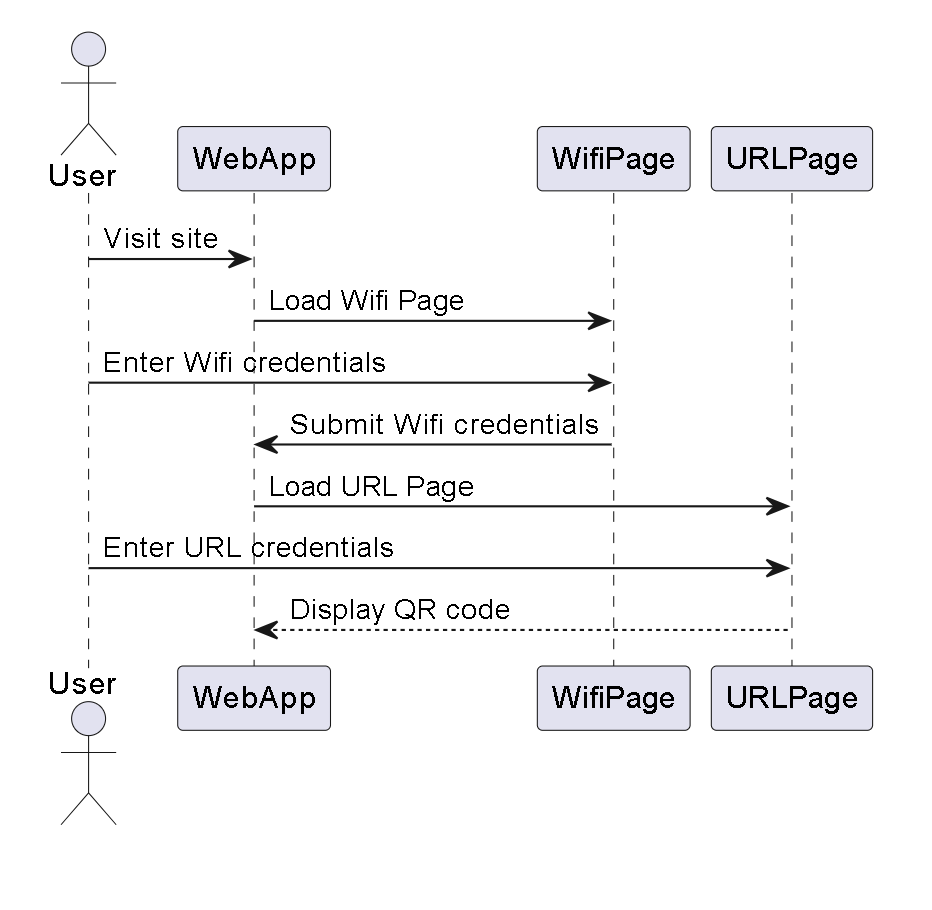
**3.4 Class Diagram**

****

*Fig 3. Class Diagram*

A class diagram is a type of static structure diagram in the Unified Modeling Language (UML) that represents the structure and behavior of a system through classes, attributes, operations, and relationships. Classes are depicted as boxes, each containing attributes (properties or variables) and operations (methods or functions). Relationships between classes, such as inheritance, association, aggregation, and composition, are represented by lines connecting the classes. Class diagrams serve as blueprints for software development, providing a visual representation of the system's architecture and helping developers understand the relationships and interactions between different components. They are essential for designing and communicating the structure of object-oriented systems, aiding in the analysis, design, and implementation phases of software development projects.

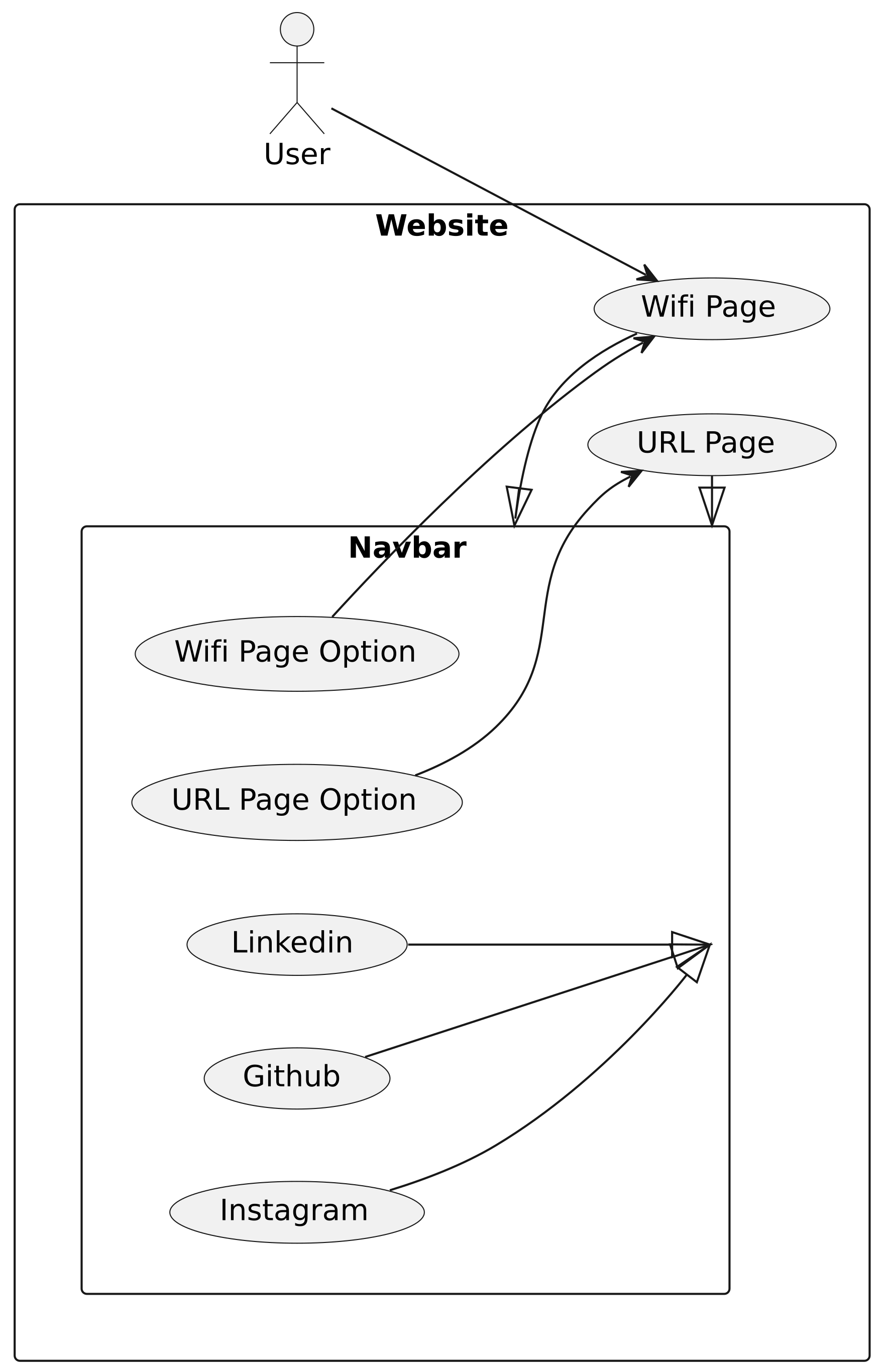
**3.5 Sequence Diagram**

****

*Fig 4. Sequence Diagram*

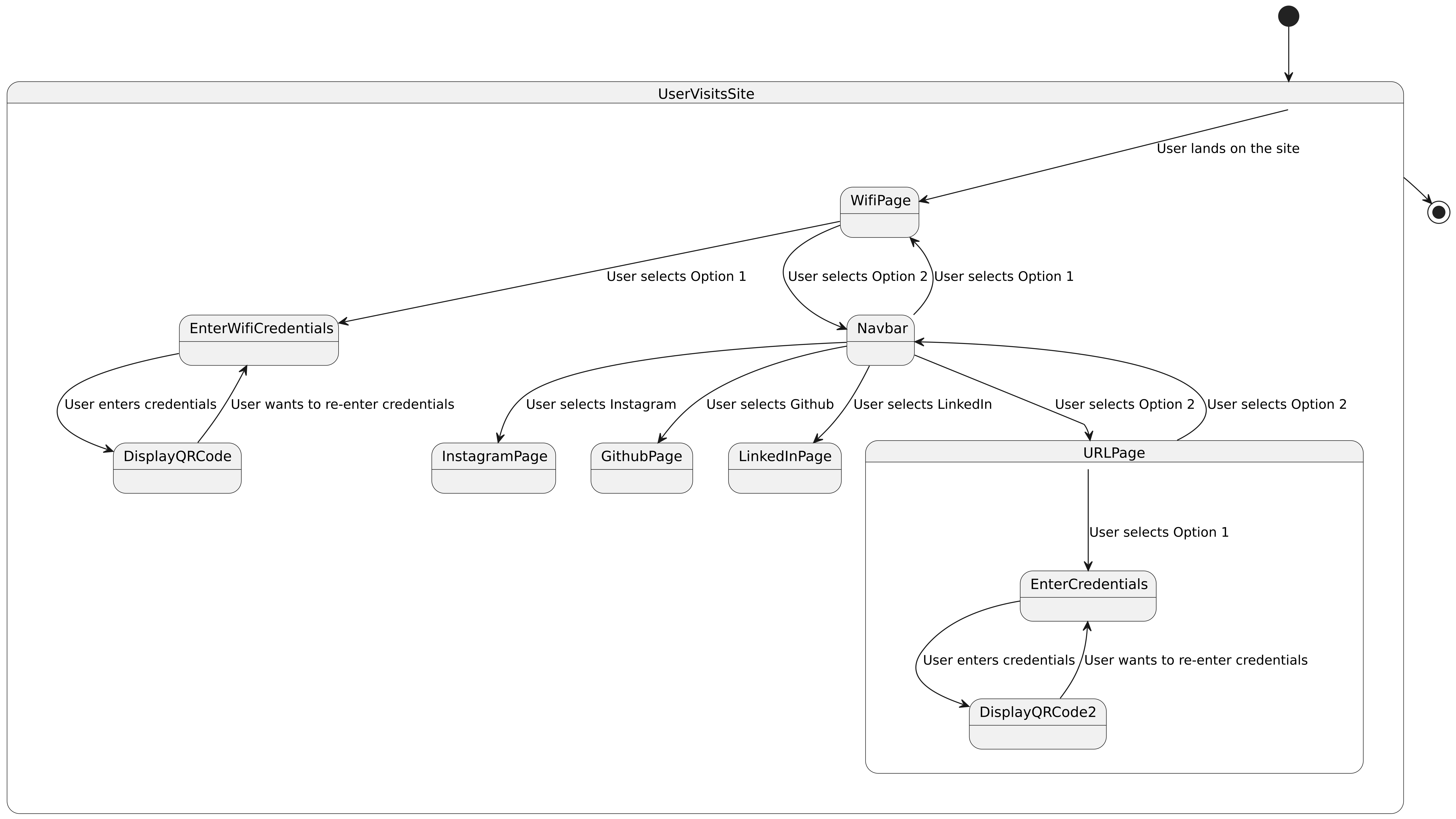
A sequence diagram is a type of interaction diagram in the Unified Modeling Language (UML) that depicts the interactions between objects or components within a system over time. It illustrates the flow of messages exchanged between these objects, showing the sequence of events that occur during the execution of a particular scenario or use case. Objects are represented as lifelines, with vertical lines indicating their existence over time, and messages between objects are depicted as horizontal arrows. Sequence diagrams are valuable tools for visualizing and understanding the dynamic behavior of a system, enabling developers to identify potential bottlenecks, analyze system behavior, and ensure that requirements are met. They are commonly used during the design and development phases of software projects to model complex interactions and verify the correctness of system behavior.

**3.6 Use Case Diagram**

****

*Fig 5. Use Case Diagram*

**3.7 State Diagram**

****

*Fig 6. State Diagram*

A state diagram models the dynamic behavior of a system by representing its states, transitions, and events. States depict various conditions or modes that the system can be in, while transitions define how the system moves between states in response to events. These diagrams are valuable for visualizing the lifecycle of objects and describing the behavior of reactive systems. State diagrams aid in understanding, analyzing, and designing complex systems, facilitating effective communication among stakeholders.

### Chapter 4

### Implementation

*This chapter includes software and hardware requirements, system coding and screen layouts.*

**4.1 System software and Hardware requirements**

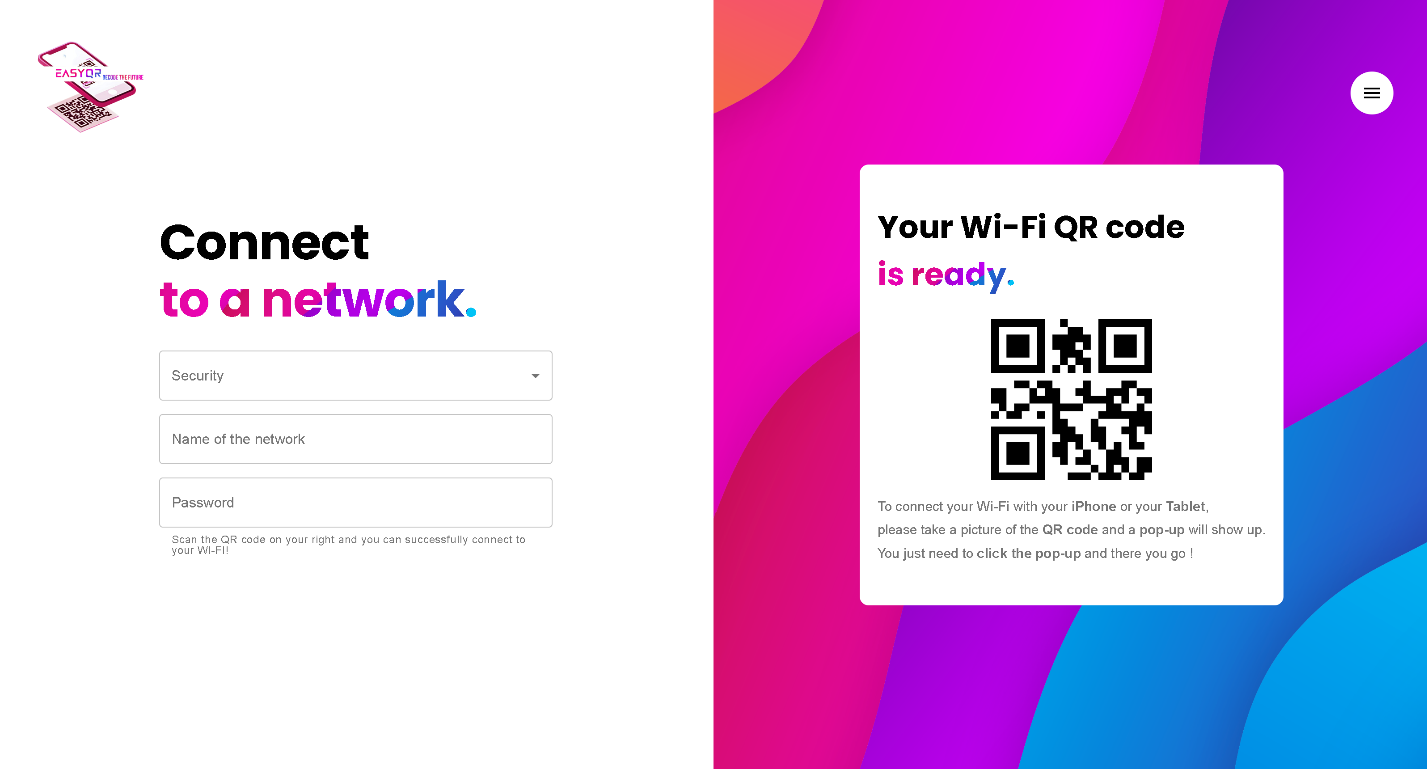
|  |  |  |
| --- | --- | --- |
| **Aspect** | **Developer Requirements** | **User Requirements** |
| **Software Requirements** | Integrated Development Environment (IDE): Visual Studio Code, Atom, WebStorm, etc. | Modern Web Browser: Google Chrome, Mozilla Firefox, Safari, etc. |
|  | Node.js and npm (Node Package Manager) or Yarn for managing dependencies | Mobile Browsers: Compatible with responsive design |
|  | Github for cloud storage of code and other modules |  |
| **Hardware Requirements** | Computer with a minimum of 4GB RAM (8GB or more recommended) and a modern multi-core processor | Devices with internet connectivity: Desktops, laptops, tablets, smartphones |
|  | Sufficient storage space for development tools, IDE, project files, and dependencies | Screen Sizes: Compatible with responsive design |
|  | Reliable internet connectivity for accessing development resources and deployment services |  |

**4.2 System Coding**

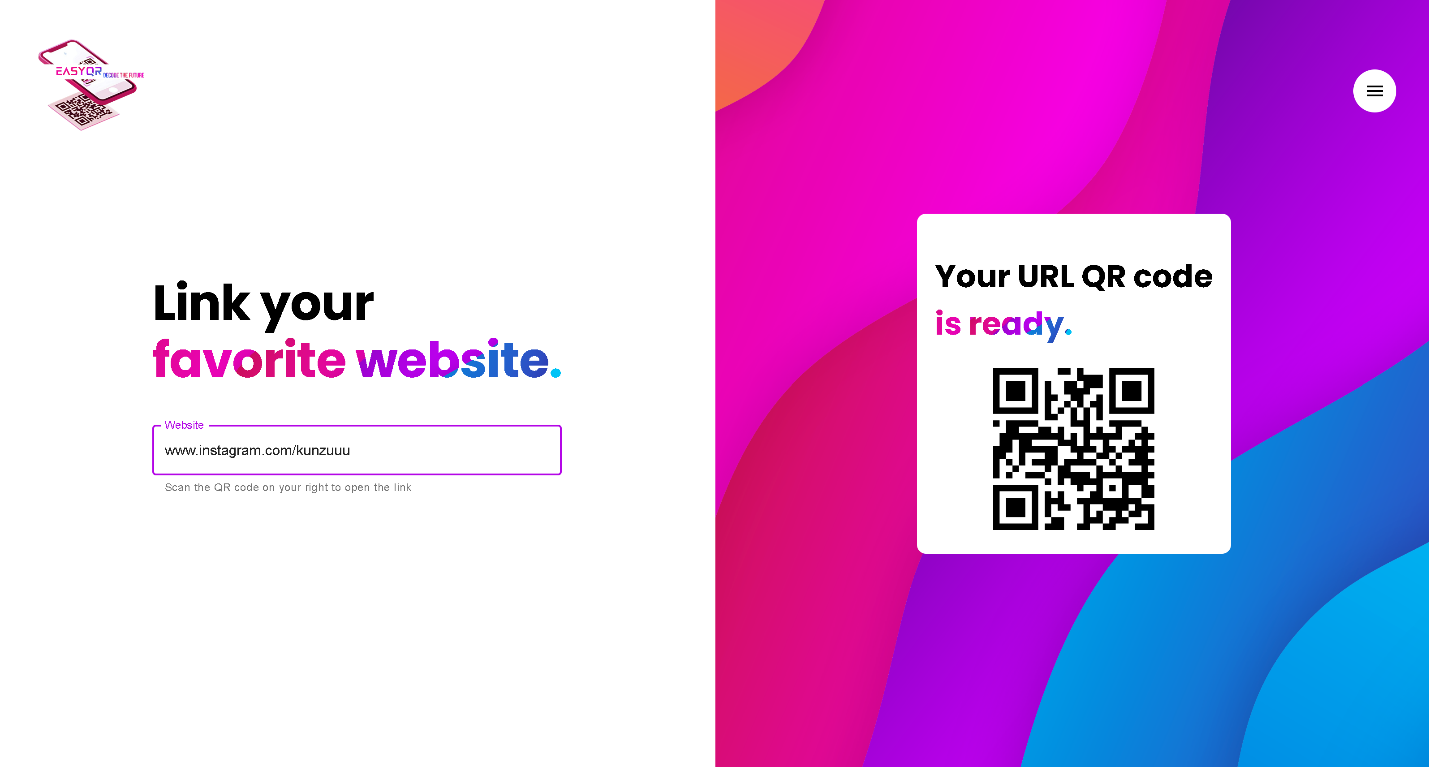
* **NavbarMenu.js**
* import React from "react";
* import { makeStyles } from "@material-ui/core/styles";
* import { Link } from "react-router-dom";
* import {
* List,
* ListItem,
* ListItemText,
* ListItemIcon,
* IconButton,
* Divider,
* Drawer
* } from "@material-ui/core";
* import InstagramIcon from '@material-ui/icons/Instagram'
* import MenuIcon from "@material-ui/icons/Menu";
* import HttpIcon from "@material-ui/icons/Http";
* import WifiIcon from "@material-ui/icons/Wifi";
* import GitHubIcon from '@material-ui/icons/GitHub'
* import LinkedInIcon from '@material-ui/icons/LinkedIn'
* const useStyles = makeStyles({
* list: {
* width: 250
* },
* fullList: {
* width: "auto"
* }
* });
* export default function TemporaryDrawer() {
* const classes = useStyles();
* const [state, setState] = React.useState({
* right: false
* });
* const toggleDrawer = (side, open) => event => {
* if (
* event.type === "keydown" &&
* (event.key === "Tab" || event.key === "Shift")
* ) {
* return;
* }
* setState({ ...state, [side]: open });
* };
* const sideList = side => (
* <div
* className={classes.list}
* role="presentation"
* onClick={toggleDrawer(side, false)}
* onKeyDown={toggleDrawer(side, false)}
* >
* <List component="nav" aria-label="main mailbox folders">
* <ListItem alignItems="flex-start">
* <img
* src="https://i.imgur.com/WEINNBb.png"
* alt="logo"
* style={{ width: 155 }}
* />
* </ListItem>
* <Divider />
* <Link to="/" style={{ textDecoration: "none", color: "black" }}>
* <ListItem button>
* <ListItemIcon>
* <WifiIcon />
* </ListItemIcon>
* <ListItemText primary="Wi-Fi" />
* </ListItem>
* </Link>
* <Link to="/url" style={{ textDecoration: "none", color: "black" }}>
* <ListItem button>
* <ListItemIcon>
* <HttpIcon />
* </ListItemIcon>
* <ListItemText primary="URL" />
* </ListItem>
* </Link>
* <a
* target="\_blank"
* rel="noopener noreferrer"
* href="https://instagram.com/kunzuuu"
* style={{ textDecoration: "none", color: "black" }}
* >
* <ListItem button>
* <ListItemIcon>
* <InstagramIcon/>
* </ListItemIcon>
* <ListItemText primary="Instagram" />
* </ListItem>
* </a>
* <a
* target="\_blank"
* rel="noopener noreferrer"
* href="https://github.com/Kunzuuu"
* style={{ textDecoration: "none", color: "black" }}
* >
* <ListItem button>
* <ListItemIcon>
* <GitHubIcon/>
* </ListItemIcon>
* <ListItemText primary="GitHub" />
* </ListItem>
* </a>
* <a
* target="\_blank"
* rel="noopener noreferrer"
* href="https://www.linkedin.com/in/kunal-singh-parmar-5005822a8?utm\_source=share&utm\_campaign=share\_via&utm\_content=profile&utm\_medium=android\_app"
* style={{ textDecoration: "none", color: "black" }}
* >
* <ListItem button>
* <ListItemIcon>
* <LinkedInIcon/>
* </ListItemIcon>
* <ListItemText primary="LinkedIn" />
* </ListItem>
* </a>
* </List>
* </div>
* );
* return (
* <div>
* <IconButton
* onClick={toggleDrawer("right", true)}
* style={{ backgroundColor: "white", fontWeight: "bold" }}
* >
* <MenuIcon style={{ color: "black" }} />
* </IconButton>
* <Drawer
* anchor="right"
* open={state.right}
* onClose={toggleDrawer("right", false)}
* >
* {sideList("right")}
* </Drawer>
* </div>
* );
* }
* **Navbar.js**
* import React from "react";
* import { Box } from "@material-ui/core";
* import { Link } from "react-router-dom";
* import NavbarMenu from "./NavbarMenu";
* const Navbar = () => {
* return (
* <div style={{ width: "100%", position: "absolute" }}>
* <Box display="flex" p={4} alignItems="center">
* <Box p={1} flexGrow={1}>
* <Link to="/" style={{ textDecoration: "none", color: "black" }}>
* <img
* src="https://i.imgur.com/WEINNBb.png"
* alt="logo"
* style={{ width: 125 }}
* />
* </Link>
* </Box>
* <Box p={1}>
* <NavbarMenu />
* </Box>
* </Box>
* </div>
* );
* };
* export default Navbar;
* **Wifi.js**
* import React, { useState } from "react";
* import {
* Grid,
* TextField,
* Typography,
* Box,
* FormControl,
* Select,
* MenuItem,
* InputLabel
* } from "@material-ui/core";
* import QRCode from 'qrcode.react';
* import Navbar from "../components/Navbar";
* const QRCodeGenerator = () => {
* const [ssid, setSsid] = useState("");
* const [ssid\_password, setSsidPassword] = useState("");
* const [security, setSecurity] = React.useState("");
* const [open, setOpen] = React.useState(false);
* const [qrCodeValue, setQrCodeValue] = useState("");
* const inputLabel = React.useRef(null);
* const [labelWidth, setLabelWidth] = React.useState(0);
* React.useEffect(() => {
* setLabelWidth(inputLabel.current.offsetWidth);
* }, []);
* const handleChange = event => {
* setSecurity(event.target.value);
* updateQrCode();
* };
* const handleClose = () => {
* setOpen(false);
* };
* const handleOpen = () => {
* setOpen(true);
* };
* const updateQrCode = () => {
* const qrCodeValue = `WIFI:T:${security};S:${ssid};P:${ssid\_password};;`;
* setQrCodeValue(qrCodeValue);
* };
* return (
* <>
* <Navbar />
* <Grid container>
* <Grid item xs={12} xl={6} md={6} sm={12}>
* <Box
* style={{ height: "100vh" }}
* display="flex"
* alignItems="center"
* justifyContent="center"
* >
* <Box style={{ padding: 25, width: "55%" }}>
* <Typography
* variant="h1"
* style={{ fontWeight: "bolder", fontSize: 55, marginBottom: 25 }}
* >
* Connect <br />
* <strong className="text\_image">to  a  network.</strong>
* </Typography>
* <FormControl fullWidth variant="outlined">
* <InputLabel ref={inputLabel} htmlFor="security-select">
* Security
* </InputLabel>
* <Select
* labelWidth={labelWidth}
* fullWidth
* open={open}
* onClose={handleClose}
* onOpen={handleOpen}
* value={security}
* onChange={handleChange}
* >
* <MenuItem value="">
* <em>None</em>
* </MenuItem>
* <MenuItem value="WEP">WEP</MenuItem>
* <MenuItem value="WPA">WPA / WPA2</MenuItem>
* </Select>
* </FormControl>
* <TextField
* style={{ marginTop: 15 }}
* variant="outlined"
* label="Name of the network"
* onChange={e => {
* setSsid(e.target.value);
* updateQrCode();
* }}
* placeholder="Wifi - Salon"
* fullWidth
* />
* <TextField
* style={{ marginTop: 15 }}
* variant="outlined"
* label="Password"
* onChange={e => {
* setSsidPassword(e.target.value);
* updateQrCode();
* }}
* placeholder="\*\*\*\*\*\*\*\*\*\*\*"
* fullWidth
* helperText="Scan the QR code on your right and you can successfully connect to your WI-FI!"
* />
* </Box>
* </Box>
* </Grid>
* <Grid item xs={12} xl={6} md={6} sm={12} className="backgroundRight">
* <Box
* style={{ height: "100vh" }}
* display="flex"
* alignItems="center"
* justifyContent="center"
* className="borderBox"
* >
* <Box
* onClick={() => {
* function generateImage(qrCodeValue) {
* console.log("QR generated")
* const win = window.open();
* win.document.write(
* '<img src="' +
* qrCodeValue +
* '" frameborder="0" style="border:0; top:0px; left:0px; bottom:0px; right:0px; width:auto; height:auto;" allowfullscreen></img>'
* );
* console.log("QR generated end")
* }
* generateImage(qrCodeValue);
* }}
* id="makePdf"
* className="hoverCard"
* style={{ backgroundColor: "white", height: "auto", padding: 20 }}
* >
* <h1 style={{ fontWeight: "bolder", fontSize: 35 }}>
* Your Wi-Fi QR code <br />
* <strong
* className="text\_image"
* style={{ backgroundPosition: "left" }}
* >
* is ready.
* </strong>
* </h1>
* <Box align="center">
* <QRCode value={qrCodeValue} size={180} fgColor="#000000" bgColor="#ffffff" />
* </Box>
* <Box style={{ marginTop: 10, marginBottom: 25 }}>
* <Typography
* variant="subtitle1"
* style={{ marginTop: 10, fontSize: 15 }}
* color="textSecondary"
* >
* To connect your Wi-Fi with your <strong>iPhone</strong> or
* your <strong>Tablet</strong>, <br />
* please take a picture of the <strong>QR code</strong> and a{" "}
* <strong>pop-up</strong> will show up. <br />
* You just need to <strong>click the pop-up</strong> and there
* you go !{" "}
* </Typography>
* </Box>
* </Box>
* </Box>
* </Grid>
* </Grid>
* </>
* );
* }
* export default QRCodeGenerator;
* **URL.js**
* import React, { useState } from "react";
* import { Grid, TextField, Typography, Box } from "@material-ui/core";
* import QRCode from 'qrcode.react';
* import Navbar from "../components/Navbar";
* const QRCodeGenerator = () => {
* const [message, setMessage] = useState("");
* const handleChange = (event) => {
* setMessage(event.target.value.trim());
* };
* const qrCodeValue = message.trim();
* return (
* <>
* <Navbar />
* <Grid container>
* <Grid item xs={12} xl={6} md={6} sm={12}>
* <Box
* style={{ height: "100vh" }}
* display="flex"
* alignItems="center"
* justifyContent="center"
* >
* <Box style={{ padding: 25 }}>
* <Typography
* variant="h1"
* style={{ fontWeight: "bolder", fontSize: 55, marginBottom: 25 }}
* >
* Link your
* <br />
* <strong className="text\_image">favorite website.</strong>
* </Typography>
* <TextField
* style={{ marginTop: 15 }}
* variant="outlined"
* label="Website"
* onChange={handleChange}
* placeholder="https://instagram.com/kunzuuu"
* fullWidth
* helperText="Scan the QR code on your right to open the link"
* inputProps={{
* style: { paddingRight: 40 }
* }}
* />
* </Box>
* </Box>
* </Grid>
* <Grid item xs={12} xl={6} md={6} sm={12} className="backgroundRight">
* <Box
* style={{ height: "100vh" }}
* display="flex"
* alignItems="center"
* justifyContent="center"
* className="borderBox"
* >
* <Box
* onClick={() => {
* const win = window.open();
* win.document.write(
* '<img src="' +
* qrCodeValue +
* '" frameborder="0" style="border:0; top:0px; left:0px; bottom:0px; right:0px; width:auto; height:auto;" allowfullscreen></img>'
* );
* }}
* id="makePdf"
* className="hoverCard"
* style={{ backgroundColor: "white", height: "auto", padding: 20 }}
* >
* <h1 style={{ fontWeight: "bolder", fontSize: 35 }}>
* Your URL QR code <br />
* <strong
* className="text\_image"
* style={{ backgroundPosition: "left" }}
* >
* is ready.
* </strong>
* </h1>
* <Box align="center">
* <QRCode value={qrCodeValue} size={180} fgColor="#000000" bgColor="#ffffff" />
* </Box>
* </Box>
* </Box>
* </Grid>
* </Grid>
* </>
* );
* };
* export default QRCodeGenerator;
* **index.js**
* import React from "react";
* import { BrowserRouter, Route, Switch } from "react-router-dom";
* import ReactDOM from "react-dom";
* import Wifi from "./pages/Wifi";
* import URL\_code from "./pages/URL";
* import "./styles.css";
* const App = () => {
* return (
* <>
* <Switch>
* <Route component={Wifi} exact path="/" />
* <Route component={URL\_code} exact path="\*" />
* </Switch>
* </>
* );
* };
* const rootElement = document.getElementById("root");
* ReactDOM.render(
* <BrowserRouter>
* <App />
* </BrowserRouter>,
* rootElement
* );
* **Styles.css**
* body {
* font-family: "Poppins", Helvetica, sans-serif !important;
* margin: 0px !important;
* }
* .MuiTypography-h1 {
* font-family: "Poppins", "Helvetica", "Arial", sans-serif !important;
* }
* .backgroundRight {
* background-image: url("https://i.imgur.com/X6kzOYN.png");
* background-repeat: no-repeat;
* background-size: cover;
* background-position: center;
* }
* .MuiOutlinedInput-root.Mui-focused .MuiOutlinedInput-notchedOutline {
* border-color: #b406e7 !important;
* }
* .MuiFormLabel-root.Mui-focused {
* color: #b406e7 !important;
* }
* .hoverCard {
* transition: all 0.3s ease;
* border-radius: 10px;
* cursor: pointer;
* }
* .hoverCard:hover {
* transform: translateY(-5px);
* box-shadow: 0px 10px 45px -6px rgba(0, 0, 0, 0.25) !important;
* }
* .nav\_text {
* font-weight: 700;
* color: white;
* margin-right: 10px !important;
* }
* .text\_image {
* color: transparent;
* background-image: url("https://i.imgur.com/X6kzOYN.png") !important;
* background-position: 50%;
* background-size: inherit;
* background-clip: text;
* -webkit-background-clip: text;
* background-repeat: no-repeat;
* background-size: cover;
* }
* .borderBox {
* -webkit-border-top-left-radius: 20px;
* -webkit-border-bottom-left-radius: 20px;
* -moz-border-radius-topleft: 20px;
* -moz-border-radius-bottomleft: 20px;
* border-top-left-radius: 20px;
* border-bottom-left-radius: 20px;
* }
* **Index.html**
* <!DOCTYPE html>
* <html lang="en">
* <head>
* <meta charset="utf-8" />
* <meta
* name="viewport"
* content="width=device-width, initial-scale=1, shrink-to-fit=no"
* />
* <meta name="theme-color" content="#000000" />
* <link
* href="https://fonts.googleapis.com/css?family=Poppins:400,700&display=swap"
* rel="stylesheet"
* />
* <link rel="shortcut icon" href="/src/assets/favicon.ico" />
* <title>Easy QR</title>
* </head>
* <body>
* <noscript>
* You need to enable JavaScript to run this app.
* </noscript>
* <div id="root"></div>
* </body>
* </html>

**4.3 Screen Layouts**

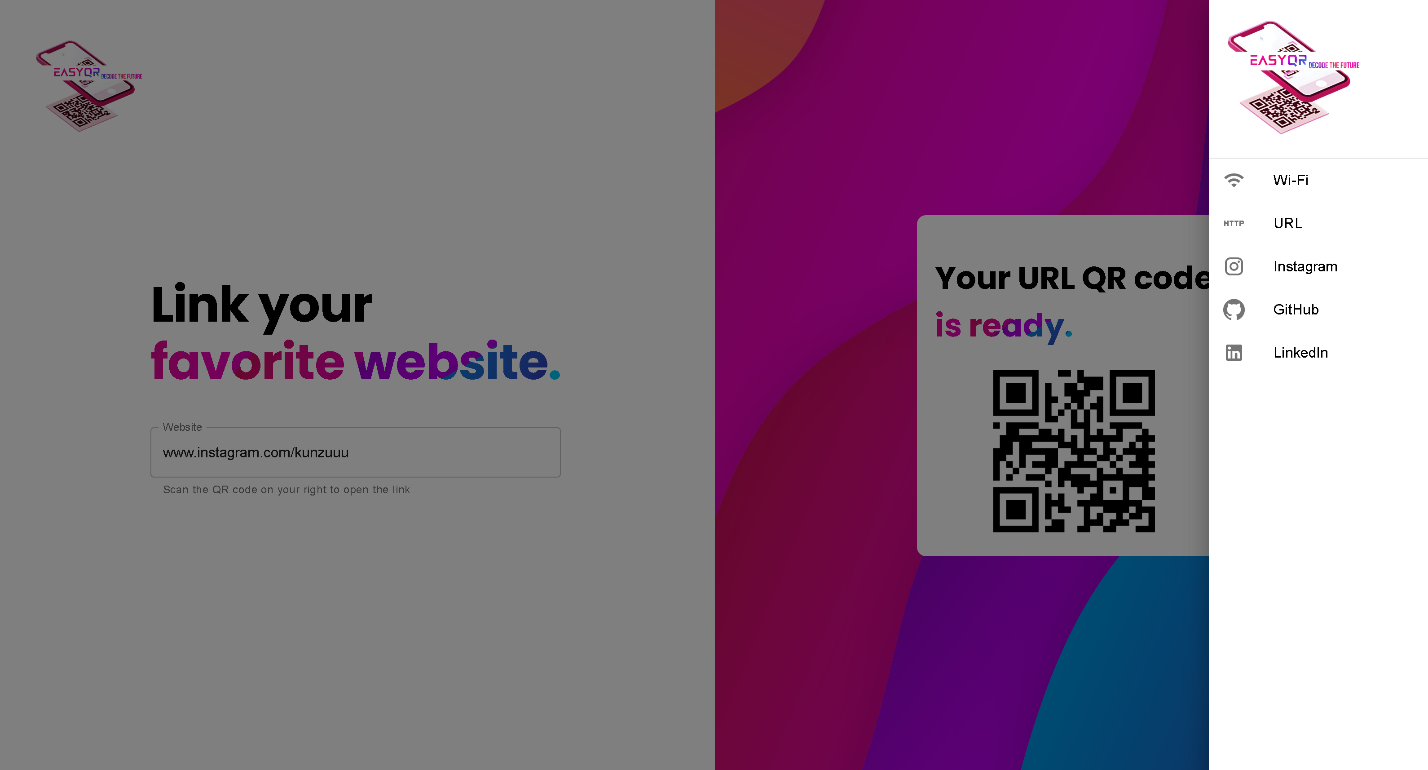
* **Wifi Page Layout**

****

* **URL Page Layout**

****

* **Navbar**

****

**Chapter 5**

**Implementation and Testing**

*This chapter includes the conclusion and future work.*

**5.1 Conclusion**

* In conclusion, this documentation has provided a comprehensive overview of our web application, showcasing its features, functionalities, and capabilities. By leveraging technologies such as JavaScript, HTML, CSS, React libraries, and Material UI libraries, we have developed a versatile tool that simplifies the process of connecting to Wi-Fi networks and accessing specific URLs with ease.
* Through intuitive design and robust functionality, our application offers users a seamless experience, allowing them to configure Wi-Fi credentials and generate QR codes effortlessly. Additionally, the integration of navigation options such as Instagram, GitHub, and LinkedIn enhances the user experience, providing convenient access to social media platforms.
* With a focus on user-centric design, security, and continuous improvement, our web application aims to meet the diverse needs of users while ensuring reliability, scalability, and performance. We believe that our application will empower users to streamline their digital interactions, enhance productivity, and enjoy seamless connectivity in an increasingly interconnected world.

**5.2 Future Work**

* Enhanced Security Features:

Implement additional security measures to further safeguard user data, such as encryption for stored credentials and secure transmission protocols.

Integrate multi-factor authentication options to enhance the authentication process and protect user accounts from unauthorized access.

Advanced User Management:

Develop features for user registration and authentication, allowing users to create accounts and personalize their experience.

Implement user profiles and settings to enable users to manage their credentials, preferences, and account details.

* Expanded QR Code Functionality:

Introduce support for additional types of QR codes, such as vCard for contact information, calendar events, or geo-location data.

Enhance QR code customization options, allowing users to adjust size, colors, and error correction levels.

* Integration with Third-Party Services:

Explore opportunities for integration with popular Wi-Fi management platforms to provide seamless connectivity experiences in various environments.

Investigate integration with URL shortening services to simplify the generation of QR codes for lengthy URLs.

* Cross-Platform Compatibility:

Optimize the web application for compatibility across a wide range of devices and browsers, ensuring consistent performance and user experience.

Develop companion mobile applications for iOS and Android platforms to complement the web-based functionality and provide a seamless mobile experience.

* User Feedback and Iterative Improvements:

Establish channels for collecting user feedback and insights to identify pain points, address usability issues, and prioritize feature requests.

Embrace an iterative development approach, releasing regular updates and improvements based on user feedback and emerging trends in web technology.

* Performance Optimization:

Conduct performance optimization efforts to improve page load times, reduce latency, and enhance responsiveness across different network conditions.



 Somaiya Vidyavihar University Vidyavihar East,   
 400077, India W: [www.somaiya.edu](http://www.somaiya.edu/)