

# **Smart Tourism**

Smart Tourism is a major project that involves creating a tourism website. The website aims to provide information about various tourist destinations, attractions, accommodations, and activities. It also includes features such as travel planning tools, booking options, interactive maps, and user reviews. The goal of the project is to enhance the overall tourism experience by providing a comprehensive and user-friendly platform for travelers.

## **● Contents:**

- |   |    |
|---|----|
| 1. Source Code of the Project .....     | 2  |
| 2. Screenshot of Running Program.....   | 23 |
| 3. User Manual for Running Program..... | 25 |

## **● Partial Source Code of the Project:**

### **smart-tour-main (main folder)**

#### **index.js**

```
const express = require("express");
const app = express();
const dotenv = require("dotenv");
const cors = require("cors");
const mongoose = require("mongoose");
const path = require("path");
const blogRouter = require("./Routes/BlogRoutes");
const reviewRouter = require("./Routes/ReviewRoutes");
const BusinessRouter = require("./Routes/BusinessRoutes");
const ContactUsRoutes = require("./Routes/ContactUsRoutes");
const ItineraryRoutes = require("./Routes/Itinerary");
const LodgePgRoutes = require("./Routes/LodgePg");
const MonumentsRoutes = require("./Routes/Monuments");
const PlacesRouter = require("./Routes/PlacesRoutes");
const UserRouter = require("./Routes/UserRoutes");
const VehicleRentalRoutes = require("./Routes/VehicleRental");
dotenv.config();
app.use(cors());
app.use(express.json());
app.use(express.static(path.join(__dirname, './frontend/build')));
const port = process.env.PORT || 5000;
mongoose.set("strictQuery", false);
const connectDb = async () => {
  try {
    await mongoose.connect(process.env.MONGODB_URL, {
```

```
useNewUrlParser: true,
useUnifiedTopology: true,
});

console.log("DB Server got connected");

} catch (err) {
  console.log(`DB Server got error ${err.message}`);
}

};

connectDb();

app.use("/blogs", blogRouter);

app.use("/reviews", reviewRouter);

app.use("/business", BusinessRouter);

app.use("/contactus", ContactUsRoutes);

app.use("/itinerary", ItineraryRoutes);

app.use("/lodgepg", LodgePgRoutes);

app.use("/monument", MonumentsRoutes);

app.use("/places", PlacesRouter);

app.use("/user", UserRouter);

app.use("/vehicle", VehicleRentalRoutes);

// app.get("/", (req, res) => {
//   res.send("App is running");
// });

app.get('*', (req, res) => {
  res.sendFile(path.resolve(__dirname, './frontend/build/index.html'));
});

const server = app.listen(port, () => {
  console.log(`App is listening on port no ${port}`);
});
```

## **FRONTEND** (sub folder)

### **Homepage.js**

```
import { useEffect, useState, useRef } from "react";
import axios from "axios";
import AOS from "aos";
import "aos/dist/aos.css";
import Isotope from "isotope-layout";
// import './App.css'
// import placesData from "places.json";
function Homepage() {
  AOS.init();
  const scriptURL =
"https://script.google.com/macros/s/AKfycbwp4Or1Hoa4VI91Nc8SBskX45a2U3e7PuyZRLWYZQe8hLcrOlqkFbi38Lk-_qlucL1H/exec";
  const form = document.forms["contactform"];
  const corporateFilterRef = useRef(null);
  const [formData, setFormData] = useState({
    companyName: "",
    ownerFullName: "",
    email: "",
    contactNumber: "",
    businessCategory: "",
    message: ""
  });
  const handleChange = (e) => {
    const { name, value } = e.target;
    setFormData({ ...formData, [name]: value });
  };
  const handleSubmit = async (e) => {
    e.preventDefault();
```

```
try {
    await axios.post(
        process.env.REACT_APP_BACKEND + "/contactus/contactus",
        formData
    );
    console.log("Form submitted successfully");
    // alert("success")
    setFormData({
        companyName: "",
        ownerFullName: "",
        email: "",
        contactNumber: "",
        businessCategory: "",
        message: ""
    });
    alert("Form Submitted")
    // You can add additional logic here after form submission if needed
} catch (error) {
    console.error("Failed to submit form:", error);
}
};

// useEffect to auto-click on .corporate after the component mounts
useEffect(() => {
    // Check if the ref is not null before attempting to click
    if (corporateFilterRef.current) {
        corporateFilterRef.current.click();
    }
}, []); // The empty dependency array ensures this effect runs only once after the component mounts
```

```
// useEffect to delay Isotope initialization

useEffect(() => {

  // Delay initialization of Isotope after rendering other sections
  const timeoutId = setTimeout(() => {

    initializeIsotope();

  }, 100); // Adjust the delay time as needed

  // Cleanup function to clear the timeout if component unmounts
  return () => clearTimeout(timeoutId);

}, []);

function onSubmit(e) {

  e.preventDefault();

  fetch(scriptURL, { method: "POST", body: new FormData(form) })

  .then((response) =>

    alert("Thank you! your form is submitted successfully.")

  )

  .then(() => {

    window.location.reload();

  })

  .catch((error) => console.error("Error!", error.message));

}

let [data, setData] = useState([]);

useEffect(() => {

  // console.log(placesData);

  // setData(placesData);

  fetchPlaces();

}, []);

// Function to initialize Isotope

function initializeIsotope() {

  // Assuming you have a grid element with class 'grid'
```

```
var grid = document.querySelector(".grid");
var gridItems = grid.querySelectorAll(".all");
var isotope = new Isotope(grid, {
  itemSelector: ".all",
  percentPosition: true,
  masonry: {
    columnWidth: ".all",
  },
});
// Assuming you have a default filter value
var defaultFilter = "*";
// Filter the grid on initialization with the default filter
isotope.arrange({ filter: defaultFilter });
// Add click event to filter items when a filter button is clicked
var filterButtons = document.querySelectorAll(".filters ul li");
filterButtons.forEach(function (button) {
  button.addEventListener("click", function () {
    filterButtons.forEach(function (btn) {
      btn.classList.remove("active");
    });
    this.classList.add("active");
    var data = this.getAttribute("data-filter");
    isotope.arrange({ filter: data });
  });
});
});
}
const fetchPlaces = async () => {
try {
  const response = await axios.get(
```

```
`${process.env.REACT_APP_BACKEND}/places/places`  
);  
  
// setData(response.data.data)  
console.log("places", response.data.data);  
  
setData(response.data.data);  
  
setDisplayedItems(response.data.data.slice(0, 5))  
  
initializeIsotope();  
  
// Check if the element with id "portfolio" exists  
  
var portfolioElement = document.getElementById("portfolio");  
  
if (portfolioElement) {  
  
    initializeIsotope();  
  
}  
  
return response.data;  
} catch (error) {  
  
    console.error("Error fetching Itinerary:", error);  
  
    return null; // Handle the error as per your application's requirement  
  
}  
};  
  
const [displayedItems, setDisplayedItems] = useState([]);  
  
const loadMoreItems = () => {  
  
    const currentItemsCount = displayedItems.length;  
  
    const nextItems = data.slice(currentItemsCount, currentItemsCount + 10);  
  
    setDisplayedItems((prevItems) => [...prevItems, ...nextItems]);  
};  
  
return (  
    <>  
    /* start banner Area */  
  
    <section className="banner-area relative" id="home">  
        <div className="container-fluid p-0">
```

```
<div
  id="carouselExampleSlidesOnly"
  className="carousel slide"
  data-bs-ride="carousel">
</div>
<div className="carousel-inner">
  <div className="carousel-item active">
    
    <div className="overlay overlay-bg"></div>
    <div
      className="banner-content carousel-caption absolute"
      data-aos="fade-down"
      data-aos-offset="200"
      data-aos-delay="20"
      data-aos-duration="2000"
      data-aos-easing="ease-in">
      <h2 className="text-uppercase text-warning">
        Land of Palaces
      </h2>
      <h1>RATNAGIRI</h1>
      <a href="/#portfolio" className="primary-btn text-uppercase">
        Explore Now
      </a>
    </div>
  </div>
</div>
```

```
<div className="carousel-item">
  
  <div className="overlay overlay-bg"></div>
  <div className="banner-content carousel-caption absolute">
    <h2 className="text-uppercase text-warning">Land of Forts</h2>
    <h1>RATNAGIRI</h1>
    <a href="/#portfolio" className="primary-btn text-uppercase">
      Explore Now
    </a>
  </div>
</div>
<div className="carousel-item">
  
  <div className="overlay overlay-bg"></div>
  <div className="banner-content carousel-caption absolute">
    <h2 className="text-uppercase text-warning">
      Land of Beaches
    </h2>
    <h1>RATNAGIRI</h1>
    <a href="/#portfolio" className="primary-btn text-uppercase">
      Explore Now
    </a>
  </div>
</div>
```

```
        </a>
    </div>
</div>
</div>
</div>
</div>
</div>
</section>
{/* End banner Area */}
{data && (
  <>
  {/* Start portfolio-area Area */}
  <section
    className="portfolio-area section-gap"
    id="portfolio"
    data-aos="fade-in"
    data-aos-offset="200"
    data-aos-delay="20"
    data-aos-duration="2000"
    data-aos-easing="ease-in"
  >
    <div className="container">
      <div className="row d-flex justify-content-center">
        <div className="menu-content col-lg-10">
          <div className="title text-center">
            <h1 className="mb-10">
              You'll fall in love with Ratnagiri
            </h1>
            <p>Explore all wonderful places in Ratnagiri</p>
          </div>
        </div>
      </div>
    </div>
  
```

```
</div>

</div>

<div className="filters">

<ul>

  <li className="active" data-filter="*">
    All
  </li>

  {/* ref={corporateFilterRef} */}

  <li data-filter=".corporate">
    Hidden Gems
  </li>

  <li data-filter=".personal">Beaches</li>
  <li data-filter=".agency">Forts</li>
  <li data-filter=".portal">Mountain Views</li>
  <li data-filter=".waterfall">Waterfalls</li>

</ul>

</div>

<div className="filters-content">

<div className="row grid">

  {/* {displayedItems?.map((item, index) => ( *)}

  {data?.map((item, index) => (
    <div
      key={index}
      className={`single-portfolio col-sm-4 all ${item.tab}`}
    >
      <div className="item">
        <img
          src={`${item.image}`}
          style={{ height: "20rem" }}
        </div>
    </div>
  ))}

</div>

<div className="grid">
  <div>
    <div>
      <img
        alt="A small thumbnail image of a landscape scene.">
    </div>
    <div>
      <h3>Hidden Gem</h3>
      <p>This is a hidden gem destination. It's a secret spot where you can experience the beauty of nature without the crowds. The area features stunning mountain views and waterfall cascades. Perfect for those who want to escape the ordinary and find something truly unique.</p>
      <button>View Details</button>
    </div>
  </div>
</div>

```

```
        alt="Work 1"
      />
      <div className="p-inner">
        <h4>
          <a href={`/ ${item.name}`}>{item.name}</a>
        </h4>
        <div className="cat">{item.location}</div>
      </div>
    </div>
  )})
</div>
</div>
/* <div className="text-center mt-4">
<button className="btn btn-primary" onClick={loadMoreItems}>
  Load More
</button>
</div> */
</div>
</section>
/* End portfolio-area Area */
</>
)
/* Start service Area */
<div id="service ">
<section
  className="service-area section-gap relative"
  // data-aos="fade-in"
  // data-aos-offset="100"
```

```
// data-aos-delay="5"
// data-aos-duration="100"
// data-aos-easing="ease-in"
>
<div className="overlay overlay-bg"></div>
<div className="container"
id ="Servicess"
>
<div className="row d-flex justify-content-center">
<div className="menu-content pb-60 col-lg-10">
<div className="title text-center">
<h1 className="mb-10 text-white">
    Always in our customer Favour
</h1>
<p className="text-white">Who are always with tight Budget</p>
</div>
</div>
</div>
</div>
</section>
{/* End service Area */}
{/* Start services Area */}
<section className="services-area pb-100">
>
<div className="container">
<div className="row">
<div
    className="col-lg-3"
    // data-aos="fade-in"
```

```
// data-aos-offset="100"
// data-aos-delay="10"
// data-aos-duration="2000"
// data-aos-easing="ease-in"

>

<div className="single-service">
  <a href="/historicplaces">
    
    <h4>Historic Monuments</h4>
    <p>
      Visit historically significant places filled with culture in
      Ratnagiri.
    </p>
  </a>
</div>
</div>

<div
  className="col-lg-3"
  // data-aos="fade-in"
  // data-aos-offset="100"
  // data-aos-delay="30"
  // data-aos-duration="2000"
  // data-aos-easing="ease-in"

>
<div className="single-service">
  <a href="/lodgepg">
    
    <h4>Hotels and Restaurants</h4>
    <p>
```

Eat rich cuisine and stay in beautiful apartments with us in  
Ratnagiri.

```
</p>
</a>
</div>
</div>
<div
  className="col-lg-3"
  // data-aos="fade-in"
  // data-aos-offset="100"
  // data-aos-delay="50"
  // data-aos-duration="2000"
  // data-aos-easing="ease-in"
>
<div className="single-service">
  <a href="/itinerary">
    
    <h4>Travel Itenaries</h4>
    <p>
      We have built perfect itenaries for you so that you dont
      have to think!!!
    </p>
  </a>
</div>
</div>
<div
  className="col-lg-3"
  // data-aos="fade-in"
  // data-aos-offset="100"
```

```
// data-aos-delay="70"
// data-aos-duration="2000"
// data-aos-easing="ease-in"
>
<div className="single-service">
  <a href="/rental">
    
    <h4>Vehicle Rentals</h4>
    <p>
      Enhance your travel experience with our convenient vehicle
      rentals tailored for tourism.
    </p>
  </a>
</div>
</div>
</div>
</div>
</section>
</div>
{/* End services Area */}
{/* Start chatbot */}
<section>
  <button className="chatbot-toggler">
    <span className="material-symbols-rounded"><i class="fa-solid fa-message"></i></span>
    <span className="material-symbols-outlined"><i class="fa-solid fa-xmark"></i></span>
  </button>
  <div className="chatbot">
    <header>
```

```
<h2 className="text-white">Smart Tourism-Chatbot</h2>
<span className="close-btn material-symbols-outlined">close</span>
</header>
<ul className="chatbox">
  <li className="chat incoming">
    <span className="material-symbols-outlined">A</span>
    <p>
      Hi there 🙌
      <br />I am Smart Tour Assistant. Ask your travel related
      queries.
    </p>
  </li>
</ul>
<div className="chat-input">
  <textarea
    placeholder="Enter a message...">
    spellCheck="false"
    required
  </textarea>
  <span id="send-btn" className="material-symbols-rounded">
    send
  </span>
</div>
</div>
</section>
/* End chatbot */
/* Start BUSINESS FORM */
<form
  method="post">
```

```
action="https://script.google.com/macros/s/AKfycbwp4Or1Hoa4VI91Nc8SBskX45a2U3e7Pu
yZRLWYZQe8hLcrOlqkFbi38Lk-_qlucL1H/exec"

onSubmit={onSubmit}

name="contactform"

id="contactform"

className="container"

// data-aos="fade-in"

// data-aos-offset="200"

// data-aos-delay="10"

// data-aos-duration="2000"

// data-aos-easing="ease-in"

>

{/* intro section */}

<div class="row">

<div class="col-md-6">

<section className="intro" id="contactus">

<h1 className="title">Ratnagiri Tourism</h1>

</section>

</div>

{/* sign-up section */}

<div class="col-md-6">

<section className="">

<p className="sign-up-para text-white font-weight-bold">

Business Enquiry Form

</p>

{/* the form itself */}

<div className="sign-up-form">

<div className="form-input">

<input
```

```
        type="text"
        // name="Company's Name"
        id="Company's Name"
        name="companyName"
        value={formData.companyName}
        onChange={handleChange}
        placeholder="Company's Name"
        required
      />
    <span>!</span>
    <p className="warning">Company's Name cannot be empty</p>
  </div>
  <div className="form-input">
    <input
      type="text"
      // name="Owner's Full Name"
      id="Owner's Full Name"
      name="ownerFullName"
      value={formData.ownerFullName}
      onChange={handleChange}
      placeholder="Owner's Full Name"
      required
    />
    <span>!</span>
    <p className="warning">"Owner's Full Name" cannot be empty</p>
  </div>
  <div className="form-input">
    <input
      type="email"
```

```
        name="email"
        id="email"
        value={formData.email}
        onChange={handleChange}
        placeholder="Email Address"
        required
      />
      <span>!</span>
      <p className="warning">Looks like this is not an email</p>
    </div>
    <div className="form-input">
      <input
        type="Contact Number"
        // name="Contact Number"
        id="Contact Number"
        name="contactNumber"
        value={formData.contactNumber}
        onChange={handleChange}
        placeholder="Contact Number"
        required
      />
      <span>!</span>
      <p className="warning">Contact Number cannot be empty</p>
    </div>
    <div className="form-input">
      <input
        type="Business Category"
        // name="Business Category"
        id="Business Category"
      />
    </div>
  </div>

```

```
        name="businessCategory"
        value={formData.businessCategory}
        onChange={handleChange}
        placeholder="Business Category"
        required
      />
      <span>!</span>
      <p className="warning">Business Category cannot be empty</p>
    </div>
    <div className="form-input">
      <input
        type="Type your message..."
        // name="Type your message..."
        id="Type your message..."
        name="message"
        value={formData.message}
        onChange={handleChange}
        placeholder="Type your message..."
        required
      />
      <span>!</span>
      <p className="warning">Type your message... cannot be empty</p>
    </div>
    <input
      className="submit-btn"
      type="submit"
      value="Submit"
      id="submit"
      onClick={handleSubmit}
    
```

```

    />

    {/* <p className="form-term">
      By clicking the button, you are agreeing to our{" "}
      <span>Terms and Services</span>{" "}
    </p> */}
    <br />
  </div>
</section>
</div>
</div>
</form>
 {/* End BUSINESS FORM */}

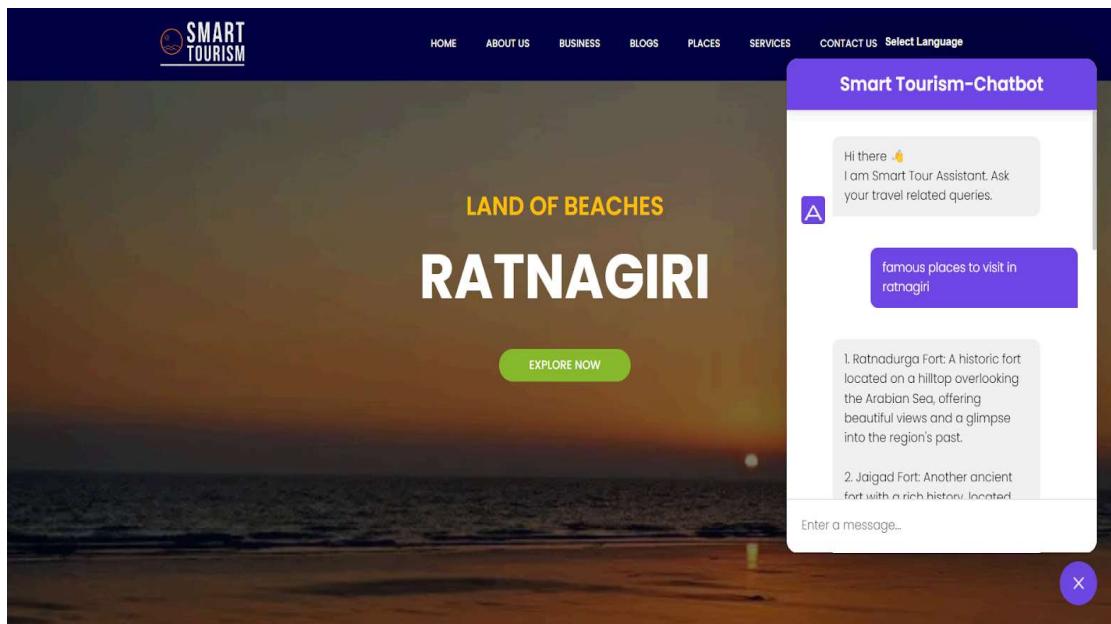
</>
);

}

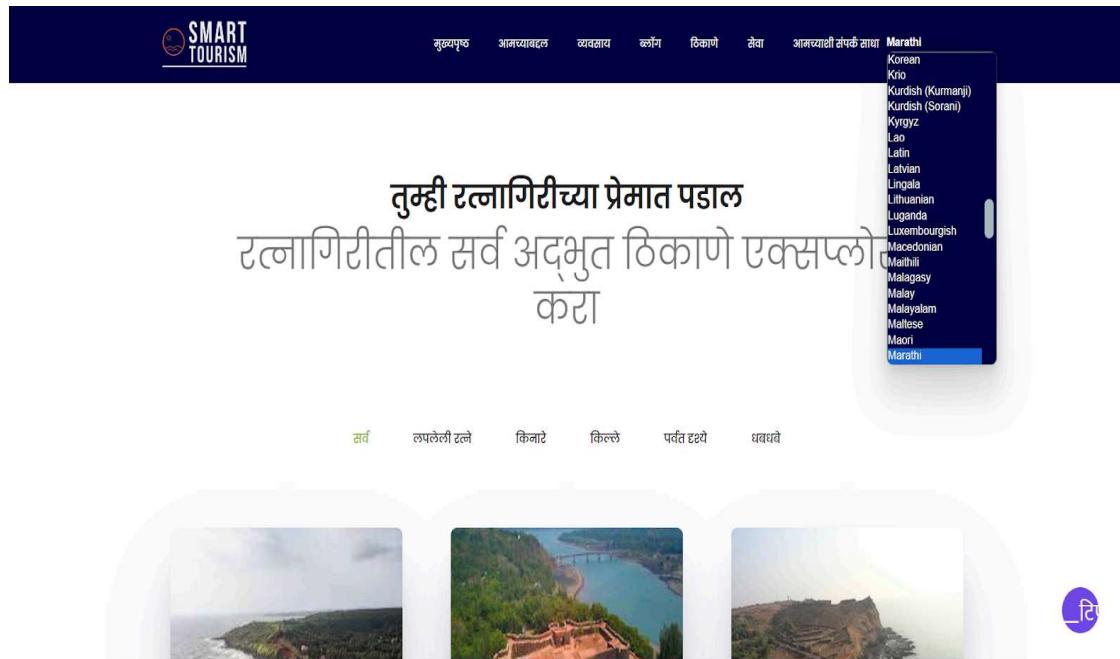
export default Homepage;

```

## Program Output

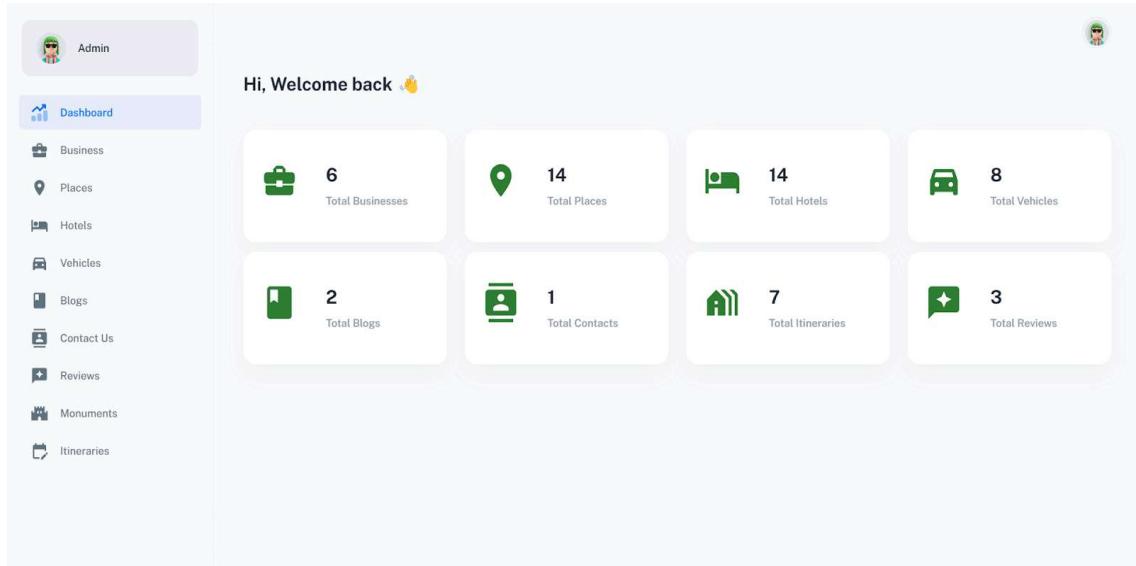


**Figure 1. Chatbot**



**Figure 2. Multilingual Support**

**Figure 3. Ratings and reviews**



**Figure 4. Dashboard**

## ● User Manual for Running the Program:

### Prerequisites:

#### **Install Packages (One-Time Setup)**

1. Navigate to the root folder: `npm install`
2. Go to the frontend folder: `npm install`
3. Proceed to the dashboard folder: `npm install`

### Steps to run the project:

To start the project, use the following command:

```
npm run dev
```

After running the project, you can access the different components at the following URLs:

1. Frontend URL: <http://localhost:3000>
2. Dashboard URL: <http://localhost:3030>
3. Backend URL: <http://localhost:5000>