Swivl - **Travel Diary Platform (Backend) Assignment**

**Submitted By**

**Manjunath Nayak**

**Nxtwave Academy**

**Step-1:** **Setup Development Environment**

**1. Install Node.js:**

**-** Visit [nodejs.org](https://nodejs.org/) and download the appropriate installer for your operating system.

- Follow the installation instructions provided for your platform. Once installed, you can verify that Node.js and npm (Node Package Manager) are correctly installed by opening a terminal or command prompt and running:

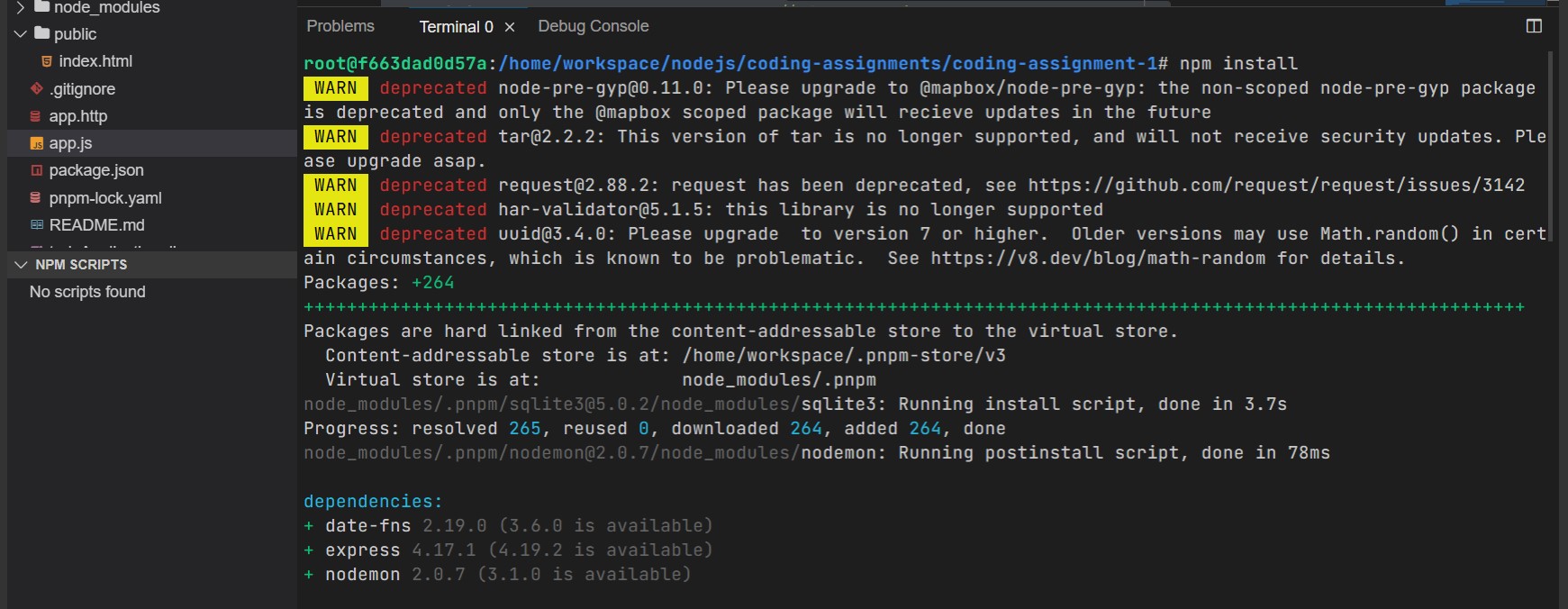
**npm install**

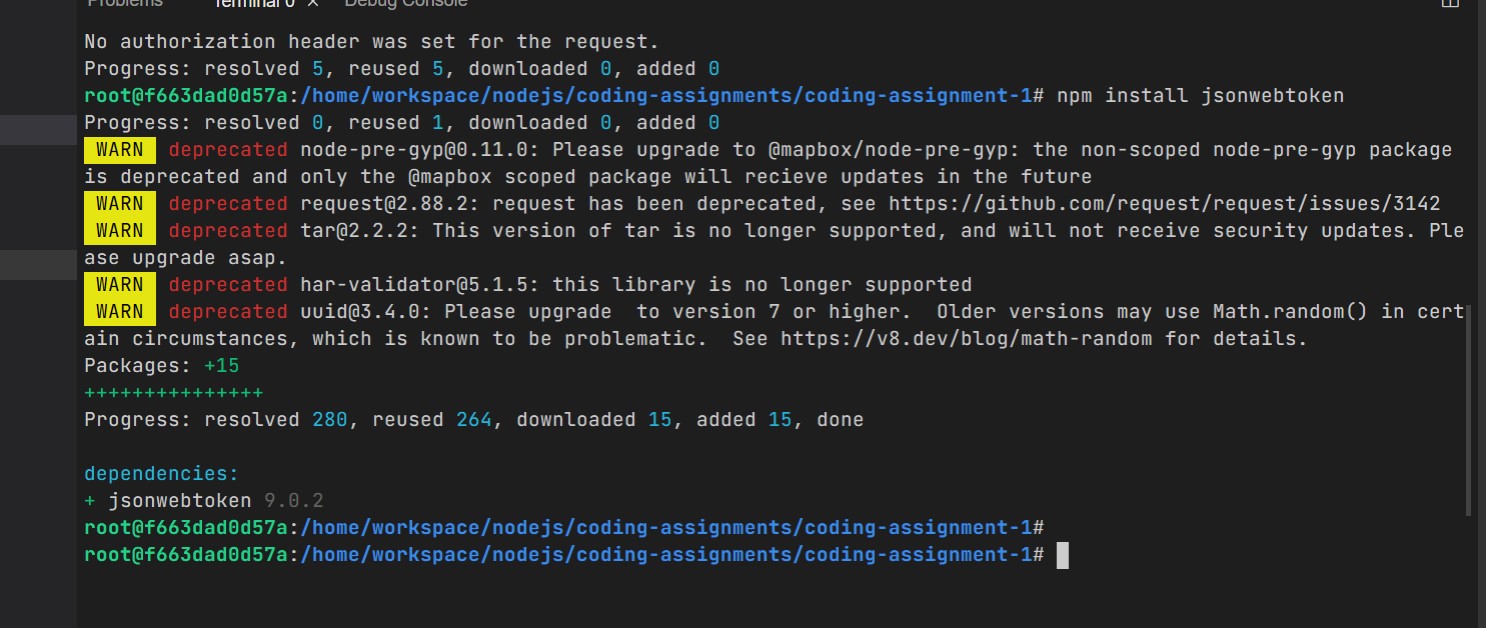
**2. Initialize Project:**

**-** Create a new directory for your project in your preferred location on your system. You can do this using the `mkdir` command in the terminal or command prompt.

- Navigate into the newly created directory using `cd`.

- Run `npm init` in the terminal or command prompt. This command will initialize a new Node.js project and prompt you to fill in details like the project name, version, description, entry point, test command, repository URL, keywords, author, and license. You can either fill in these details or press enter to accept the default values.

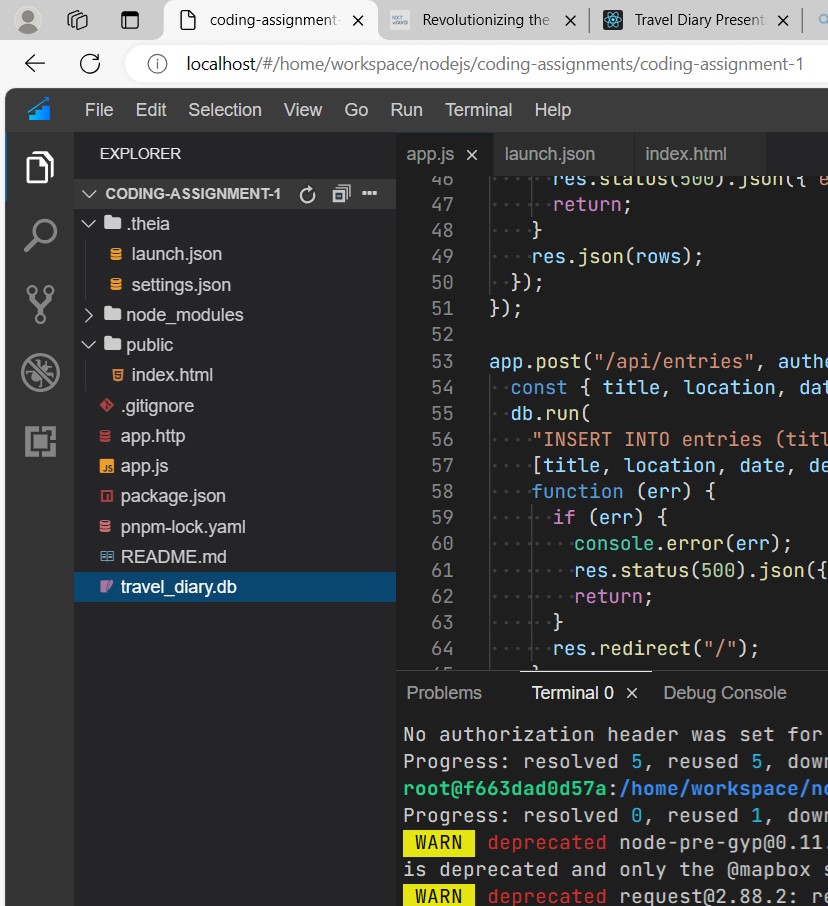


**Step 2:Installing Dependencies:**

**Step 3: Setup Basic Server**

**1.Creating App.js file**

**2.Creating Index.html**



**Step 4:-Database Setup**

* Install Database: Install your chosen database and set it up on your local machine.
* Connect Database: In your project, write code to connect to the database.
* Step : Initializing SQLite database
* const db = new sqlite3.Database("travel\_diary.db");

**Step 5**:-**Implementation Of CRUD(Create,Read,Update,Delete) Operations**

**Create:**

// Create a new travel entry

app.post("/api/entries", authenticateUser, (req, res) => {

  const { title, location, date, description } = req.body;

  db.run(

    "INSERT INTO entries (title, location, date, description) VALUES (?, ?, ?, ?)",

    [title, location, date, description],

    function (err) {

      if (err) {

        console.error(err);

        res.status(500).json({ error: "Internal Server Error" });

        return;

      }

      res.redirect("/");

    }

  );

});

**Update:**

// Update a travel entry

app.put("/api/entries/:id", authenticateUser, (req, res) => {

  const id = req.params.id;

  const { title, location, date, description } = req.body;

  db.run(

    "UPDATE entries SET title = ?, location = ?, date = ?, description = ? WHERE id = ?",

    [title, location, date, description, id],

    function (err) {

      if (err) {

        console.error(err);

        res.status(500).json({ error: "Internal Server Error" });

        return;

      }

      res.json({ message: "Entry updated successfully" });

    }

  );

});

**Delete:**

// Delete a travel entry

app.delete("/api/entries/:id", authenticateUser, (req, res) => {

  const id = req.params.id;

  db.run("DELETE FROM entries WHERE id = ?", [id], function (err) {

    if (err) {

      console.error(err);

      res.status(500).json({ error: "Internal Server Error" });

      return;

    }

    res.json({ message: "Entry deleted successfully" });

  });

});

**7. Setup Routes:**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Travel Diary Presentation</title>

    <style>

      /\* Your CSS styles here \*/

      .slides {

        background-size: cover;

        text-align: center;

      }

      .k {

        background-image: url("https://content.thriveglobal.com/wp-content/uploads/2020/10/Travel.jpg");

        background-size: cover;

        background-position: center;

        text-align: center;

      }

      /\* public/style.css \*/

      .body {

        font-family: Arial, sans-serif;

        margin: 0;

        padding: 0;

        font-family: Arial, sans-serif;

        margin: 0;

        padding: 0;

        display: flex;

        justify-content: center;

        align-items: center;

        height: 100vh;

      }

      .container {

        max-width: 800px;

        margin: 0 auto;

        padding: 20px;

      }

      h1 {

        text-align: center;

        margin-bottom: 20px;

      }

      form {

        margin-bottom: 20px;

      }

      label {

        display: block;

        margin-bottom: 5px;

      }

      input[type="text"],

      input[type="date"],

      textarea {

        width: 100%;

        padding: 10px;

        margin-bottom: 10px;

        border: 1px solid #ccc;

        border-radius: 5px;

        box-sizing: border-box;

      }

      .k {

        background-image: url("https://careymartell.com/wp-content/uploads/2015/10/Rare\_Landing\_Mountain.jpg");

        height: 100vh;

        width: 100vw;

        background-size: cover;

        background-position: center;

        text-align: center;

      }

      .o {

        color: white;

      }

      button {

        padding: 10px 20px;

        background-color: #007bff;

        color: #fff;

        border: none;

        border-radius: 5px;

        cursor: pointer;

      }

      button:hover {

        background-color: #0056b3;

      }

      ul {

        list-style-type: none;

        padding: 0;

      }

      li {

        margin-bottom: 10px;

      }

      li button {

        margin-left: 10px;

        padding: 5px 10px;

        background-color: #dc3545;

        color: #fff;

        border: none;

        border-radius: 5px;

        cursor: pointer;

      }

      li button:hover {

        background-color: #c82333;

      }

      .matter {

        display: flex;

        flex-direction: column;

        justify-content: center;

      }

    </style>

  </head>

  <body>

    <div class="k">

      <div class="slides" id="slide1">

        <h1 class="o">Travel Diary Platform</h1>

        <p class="o">

          Welcome to the Travel Diary Platform. Keep track of your adventures!

        </p>

        <button onclick="showDiary()">Use Diary</button>

      </div>

    </div>

    <div>

      <div class="slides" id="slide2">

        <!-- Insert the provided HTML code here -->

        <h1>Travel Diary</h1>

        <form id="entryForm" action="/api/entries" method="POST">

          <label for="title">Title:</label>

          <input type="text" id="title" name="title" required /><br />

          <label for="location">Location:</label>

          <input type="text" id="location" name="location" required /><br />

          <label for="date">Date:</label>

          <input type="date" id="date" name="date" required /><br />

          <label for="description">Description:</label><br />

          <textarea

            id="description"

            name="description"

            rows="4"

            cols="50"

            required

          ></textarea

          ><br />

          <button type="submit">Add Entry</button>

        </form>

        <h2>Entries:</h2>

        <ul id="entries"></ul>

        <script>

          // Your JavaScript code here

          function fetchEntries() {

            fetch("/api/entries", {

              headers: {

                Authorization: `Bearer ${localStorage.getItem("token")}`,

              },

            })

              .then((response) => response.json())

              .then((entries) => {

                const entriesList = document.getElementById("entries");

                entriesList.innerHTML = "";

                entries.forEach((entry) => {

                  const li = document.createElement("li");

                  li.textContent = `${entry.title} - ${entry.location} - ${entry.date}`;

                  const updateButton = document.createElement("button");

                  updateButton.textContent = "Update";

                  updateButton.onclick = () => {

                    updateEntry(entry.id);

                  };

                  const deleteButton = document.createElement("button");

                  deleteButton.textContent = "Delete";

                  deleteButton.onclick = () => {

                    deleteEntry(entry.id);

                  };

                  li.appendChild(updateButton);

                  li.appendChild(deleteButton);

                  entriesList.appendChild(li);

                });

              })

              .catch((error) =>

                console.error("Error fetching entries:", error)

              );

          }

          function updateEntry(id) {

            const entry = prompt(

              "Enter updated entry (title, location, date, description):"

            );

            if (entry) {

              const [title, location, date, description] = entry

                .split(",")

                .map((e) => e.trim());

              fetch(`/api/entries/${id}`, {

                method: "PUT",

                headers: {

                  "Content-Type": "application/json",

                  Authorization: `Bearer ${localStorage.getItem("token")}`,

                },

                body: JSON.stringify({ title, location, date, description }),

              })

                .then(() => fetchEntries())

                .catch((error) =>

                  console.error("Error updating entry:", error)

                );

            }

          }

          function deleteEntry(id) {

            if (confirm("Are you sure you want to delete this entry?")) {

              fetch(`/api/entries/${id}`, {

                method: "DELETE",

                headers: {

                  Authorization: `Bearer ${localStorage.getItem("token")}`,

                },

              })

                .then(() => fetchEntries())

                .catch((error) =>

                  console.error("Error deleting entry:", error)

                );

            }

          }

          document

            .getElementById("entryForm")

            .addEventListener("submit", (event) => {

              event.preventDefault();

              const formData = new FormData(event.target);

              fetch("/api/entries", {

                method: "POST",

                headers: {

                  Authorization: `Bearer ${localStorage.getItem("token")}`,

                },

                body: formData,

              })

                .then(() => fetchEntries())

                .catch((error) => console.error("Error adding entry:", error));

            });

          fetchEntries();

        </script>

      </div>

    </div>

    <script>

      let slideIndex = 1;

      showSlide(slideIndex);

      function showSlide(n) {

        const slides = document.getElementsByClassName("slides");

        if (n > slides.length) {

          slideIndex = 1;

        }

        if (n < 1) {

          slideIndex = slides.length;

        }

        for (let i = 0; i < slides.length; i++) {

          slides[i].style.display = "none";

        }

        slides[slideIndex - 1].style.display = "block";

      }

      function showDiary() {

        slideIndex = 2;

        showSlide(slideIndex);

      }

    </script>

  </body>

</html>

**STEP:Middleware for Authentication**

const express = require("express");

const sqlite3 = require("sqlite3").verbose();

const bodyParser = require("body-parser");

const jwt = require("jsonwebtoken");

const path = require("path");

const app = express();

const PORT = process.env.PORT || 3000;

const SECRET\_KEY = "your\_secret\_key"; // Change this to a secure secret key

// Initialize SQLite database

const db = new sqlite3.Database("travel\_diary.db");

// Middleware

app.use(bodyParser.urlencoded({ extended: true }));

app.use(bodyParser.json());

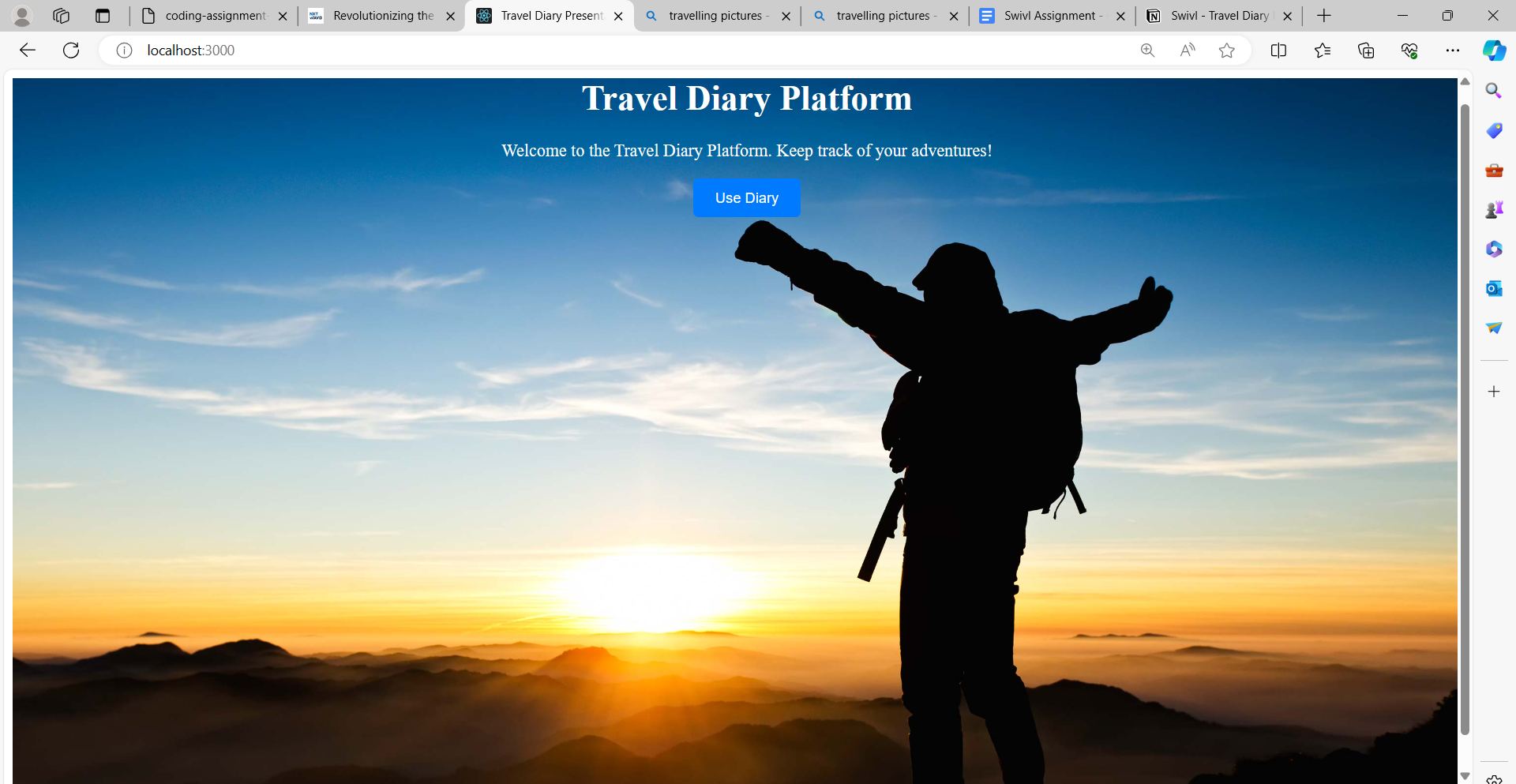
app.use(express.static(path.join(\_\_dirname, "public")));

// Generate JWT token

function generateToken(user) {

  return jwt.sign({ id: user.id }, SECRET\_KEY, { expiresIn: "1h" });

}

**Output:**

