l want to show you three seperate files we were working on, its a html, javascript, node and sql project called "interaa" that shortens users url on the same page without redirecting or reloading in the format of interaa.c/drebj as an example. I have the index.html file, the server.js and the database.js and routes.js containing mysql2 library and all dependencies installed in package.json

the mysql has been installed on my windows 11, and server is running, I have even created a database and table

index.html:

<html>

  <head>

    <title>Interaa - URL Shortener</title>

    <style>

      /\* Add CSS styles for yellow background and blue buttons here \*/

      body {

        background-color: yellow;

      }

      button {

        background-color: blue;

        color: white;

        font-size: 16px;

        border: none;

        padding: 8px 16px;

        cursor: pointer;

      }

      /\* Add CSS styles for Bungee Hairline font here \*/

      body {

        font-family: "Bungee Hairline", sans-serif;

      }

    </style>

  </head>

  <body>

    <h1>Interaa - URL Shortener</h1>

    <form id="shortener-form">

      <label for="url">Enter a URL to shorten:</label><br />

      <input type="text" id="url" name="url" /><br />

      <button type="submit">Shorten</button>

    </form>

    <script src="/server/node\_modules/validator/validator.js"></script>

    <script>

      // Function to send a request to the server to shorten a URL

      const sendShortenRequest = (url) => {

        // Send an HTTP POST request to the server to shorten the URL

        fetch("/shorten", {

          method: "POST",

          headers: { "Content-Type": "application/json" },

          body: JSON.stringify({ url: url }),

        })

          .then((response) => {

            // Check the status code of the response

            if (response.status !== 200) {

              throw new Error(`Error: ${response.status}`);

            }

            return response.json();

          })

          .then((data) => {

            // Display the short URL to the user

            alert(data.shortUrl);

          })

          .catch((error) => {

            console.error(error);

            alert("An error occurred while trying to shorten the URL");

          });

      };

      document

        .getElementById("shortener-form")

        .addEventListener("submit", function (event) {

          event.preventDefault();

          // Get the URL from the form input field

          const url = document.getElementById("url").value;

          // Validate the URL using a library like validator.js

          if (!validator.isURL(url)) {

            alert("Please enter a valid URL");

            return;

          }

          // Call the sendShortenRequest function with the URL

          sendShortenRequest(url);

        });

    </script>

  </body>

</html>

server.js:

const express = require("express");

const shortid = require("shortid");

const mysql2 = require("mysql2");

const validator = require("validator");

// Require the database connection

const database = require("./database");

// Select all rows from the database

database.selectAll((rows) => {

console.log(rows);

// Close the connection

connection.end();

});

// Configure the database connection

const connection = mysql2.createConnection({

host: "localhost",

user: "root",

port: 3300,

password: "root",

database: "shorter",

});

// Create an Express app

const app = express();

// Parse incoming request bodies as JSON

app.use(express.json());

// Create an HTTP endpoint to shorten a URL

app.post("/shorten", (req, res) => {

console.log("Received request:", req.body); // Log the request body

console.log("Request headers:", req.headers); // Log the request headers

try {

// Get the URL to shorten from the request body

const { url } = req.body;

// Validate the URL using a library like validator.js

if (!validator.isURL(url)) {

res.sendStatus(400);

return;

}

// Generate a short ID using shortid

const shortId = shortid.generate();

// Construct the short URL using the desired format

const shortUrl = `interaa.c/${shortId}`;

// Save the original URL and the short URL to the database

connection.query(

"INSERT INTO shorter (short\_id, url) VALUES (?, ?)",

[shortId, url],

(error, result) => {

if (error) throw error;

// Send the short URL as a string in the response

res.send(JSON.stringify({ shortUrl: shortUrl }));

}

);

} catch (error) {

console.error(error);

res.sendStatus(500);

}

console.log("Sending response:", res.statusCode, res.body); // Log the response

});

// Define a function to get the original URL from the database

function getOriginalUrl(shortId, callback) {

connection.query(

"SELECT url FROM links WHERE short\_url = ?",

[shortId],

(error, results) => {

if (error) {

callback(error);

return;

}

callback(null, results[0].url);

}

);

}

// Redirect short URLs to the original URL

app.get("/:shortId", (req, res) => {

try {

const shortId = req.params.shortId;

getOriginalUrl(shortId, (error, url) => {

if (error) {

console.error(error);

res.sendStatus(500);

return;

}

if (!url) {

res.sendStatus(404);

return;

}

res.redirect(url);

});

} catch (error) {

console.error(error);

res.sendStatus(500);

}

});

// Serve the index.html file for all other requests

app.get("\*", (req, res) => {

try {

res.sendFile(path.join(\_\_dirname, "index.html"));

} catch (error) {

console.error(error);

res.sendStatus(500);

}

});

// Start the server

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

app.listen(port, () => {

console.log(`Server listening on port ${port}`);

});

database.js:

const mysql2 = require("mysql2");

// Configure the database connection

const connection = mysql2.createConnection({

host: "localhost",

user: "root",

port: 3300,

password: "root",

database: "shorter",

});

const createTable = () => {

connection.query(

`CREATE TABLE links (

id INT AUTO\_INCREMENT PRIMARY KEY,

url VARCHAR(255) NOT NULL,

short\_url VARCHAR(255) NOT NULL

)`,

(error, results) => {

if (error) {

console.error(error);

return;

}

console.log("Table created successfully");

}

);

};

const selectAll = (callback) => {

// Connect to the database

connection.connect((err) => {

if (err) {

console.error(err);

return;

}

console.log("Connected to the database");

// Execute the SELECT query

connection.query("SELECT \* FROM links", (error, rows) => {

if (error) {

console.error(error);

return;

}

callback(rows);

});

});

};

// Export the createTable and selectAll functions

module.exports = { createTable, selectAll };

routes.js:

const express = require("express");

const router = express.Router();

// Define a function to get the original URL from the database

function getOriginalUrl(shortId, callback) {

connection.query(

"SELECT url FROM links WHERE short\_url = ?",

[shortId],

(error, results) => {

if (error) {

console.error(error);

callback(null);

return;

}

if (results.length > 0) {

callback(results[0].url);

} else {

callback(null);

}

}

);

}

// Create an endpoint to redirect the user to the original URL

router.get("/:shortId", (req, res) => {

// Get the short ID from the request parameters

const shortId = req.params.shortId;

// Get the original URL from the database

getOriginalUrl(shortId, (url) => {

if (url) {

// Redirect the user to the original URL

res.redirect(url);

} else {

// Send a 404 Not Found response if the short ID is invalid

res.sendStatus(404);

}

});

});

module.exports = router;

package.json:

{

"name": "server",

"version": "1.0.0",

"description": "",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"author": "",

"license": "ISC",

"dependencies": {

"mysql2": "^2.3.3",

"shortid": "^2.2.16",

"validator": "^13.7.0"

}