NTERMEDIATE Web Development Project:

Task 1 : Link Shortener

In this project, you will be required to make an API to build short URLs. The functionality will be similar to bitly. Using Node, Express, and MongoDB you can make your own URL Shortener service. However, you can use any backend language also depending on your expertise. The project is not specific to any particular backend language.

Skills Required – Node, MongoDB,JavaScript internship report documetnation

Introduction

Overview

This internship involved the development of a URL Shortener service using Node.js, Express, and MongoDB. The project aimed to replicate the functionality of popular services like Bitly.

Objectives

Create a robust backend API for shortening URLs.

Utilize Node.js for server-side logic.

Implement data storage and retrieval using MongoDB.

Develop RESTful API endpoints for URL shortening and redirection.

Technologies Used

Backend: Node.js, Express

Database: MongoDB

Additional Libraries: Mongoose (MongoDB ODM), shortid (for generating short URLs)

Project Structure

app.js: Main application file containing server setup, route definitions, and middleware.

models/: Directory containing MongoDB schema definitions.

routes/: Directory containing Express route handlers.

Implementation Details

Setting Up the Environment

Initializing the Project:

Created a new Node.js project using npm init.

Installed necessary packages (express, mongoose, shortid).

Express Server:

Established an Express server to handle HTTP requests.

Implemented middleware for parsing JSON.

Database Integration

MongoDB Connection:

Connected the application to a MongoDB database.

Utilized Mongoose for schema definition and interaction with the database.

URL Schema:

Defined a MongoDB schema for storing original URLs and their corresponding short URLs.

URL Shortening Logic

Shortening Endpoint (POST /api/shorten):

Implemented a route for shortening URLs.

Generated short URLs using the shortid library.

Saved the original and short URLs to the MongoDB database.

Redirection Endpoint (GET /:shortUrl):

Created a route for redirecting users from short URLs to their original destinations.

Retrieved the original URL from the database based on the short URL.

Challenges Faced

Validation:

Implemented validation for URL formats to ensure data integrity.

Error Handling:

Developed error-handling mechanisms for cases where short URLs are not found.

Future Improvements

Custom Aliases:

Enhance the system to allow users to set custom aliases for short URLs.

Analytics:

Integrate analytics to track the number of clicks on each short URL.

Conclusion

The internship provided valuable hands-on experience in building a practical web development project. It enhanced my skills in backend development using Node.js, working with MongoDB for data storage, and creating RESTful APIs. The challenges faced during the project contributed to my problem-solving abilities, and I look forward to applying these skills in future projects.

Acknowledgments

I would like to express my gratitude to [Company Name] for providing me with the opportunity to work on this project. Special thanks to my mentor [Mentor's Name] for guidance and support throughout the internship.