Project 1: URL Shortener Service

To build a URL shortener service using Node, Express, and MongoDB, here's a high-level overview of the steps:

Set up a new Node project and install the required dependencies: express, mongodb, and mongoose (a MongoDB ORM for Node).

Create a MongoDB database to store the short URLs and their corresponding original URLs.

Define a schema for the short URL documents using Mongoose.

Create an Express API with two endpoints:

POST /shorten: takes a URL as input and generates a short URL, storing it in the database.

GET /:shortUrl: redirects the user to the original URL corresponding to the short URL.

Implement the logic to generate a unique short URL for each input URL.

Test the API using a tool like Postman or cURL.

Here's some sample code to get you started:

// server.js

const express = require('express');

const mongoose = require('mongoose');

const app = express();

mongoose.connect('mongodb://localhost/url-shortener', { useNewUrlParser: true, useUnifiedTopology: true });

const ShortUrl = mongoose.model('ShortUrl', {

originalUrl: String,

shortUrl: String

});

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

const originalUrl = req.body.url;

const shortUrl = generateShortUrl(); // implement this function to generate a unique short URL

const shortUrlDoc = new ShortUrl({ originalUrl, shortUrl });

shortUrlDoc.save((err, doc) => {

if (err) {

res.status(500).send({ message: 'Error creating short URL' });

} else {

res.send({ shortUrl: doc.shortUrl });

}

});

});

app.get('/:shortUrl', (req, res) => {

const shortUrl = req.params.shortUrl;

ShortUrl.findOne({ shortUrl }, (err, doc) => {

if (err ||!doc) {

res.status(404).send({ message: 'Short URL not found' });

} else {

res.redirect(doc.originalUrl);

}

});

});

app.listen(3000, () => {

console.log('Server listening on port 3000');

});