1. Create a new project directory and navigate to it using the command line interface (CLI). You can do this by running the following commands:

mkdir project-name

cd project-name

1. Initialize a new Node.js project by running the following command:

npm init

This will create a **package.json** file in your project directory.

1. Install the required packages: Express.js, EJS, Express-EJS-Layout, MongoDB, and Mongoose. You can do this by running the following command:

npm install express ejs express-ejs-layout mongoose mongodb --save

1. Create a new folder called **config** in the project directory. Inside it, create a new file called **database.js**, and add the following code to it:

// Require necessary modules

const mongoose = require('mongoose');

// Connect to database

*mongoose.connect('mongodb://localhost/mydatabase', {*

*useNewUrlParser: true,*

*useUnifiedTopology: true,*

*useCreateIndex: true,*

*useFindAndModify: false*

*}).then(() => {*

*console.log('Database connected successfully');*

*}).catch((err) => {*

*console.error('Database connection error:', err);*

*});*

This code sets up a connection to a MongoDB database using Mongoose.

1. Create a new folder called **assets** in the project directory. Inside it, create a new folder called **css**, and add a new file called **style.css** to it. You can add your own CSS styles to this file.
2. Create a new folder called **views** in the project directory, and inside it, create a new file called **layout.ejs**. Add the following code to it

<!DOCTYPE html>

<html>

<head>

<title><%= title %></title>

<link rel="stylesheet" href="/assets/css/style.css">

</head>

<body>

<header>

<nav>

<ul>

<li><a href="/">Home</a></li>

<li><a href="/signup">Sign Up</a></li>

</ul>

</nav>

</header>

<main>

<%- body %>

</main>

<footer>

&copy; My Website <%= new Date().getFullYear() %>

</footer>

</body>

</html>

This code defines the HTML markup for the layout of your website, which includes a header, main content area, and footer.

1. Inside the **views** folder, create a new file called **home.ejs**, and add the following code to it:

<h1>Welcome to my website!</h1>

<p>Click <a href="/signup">here</a> to sign up.</p>

This code defines the HTML markup for the home page of your website.

1. Inside the **views** folder, create a new file called **signup.ejs**, and add the following code to it:

<h1>Sign Up</h1>

<form action="/signup" method="post">

<label for="name">Name:</label>

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

<label for="email">Email:</label>

<input type="email" id="email" name="email" required><br><br>

<label for="password">Password:</label>

<input type <password" id="password" name="password" required><br><br>

<button type="submit">Sign Up</button>

</form>

This code defines the HTML markup for the signup form of your website, which includes input fields for the user's name, email, and password.

1. Inside the project directory, create a new folder called **models**, and add a new file called **user.js** to it. Add the following code to it:

// Require necessary modules

const mongoose = require('mongoose');

// Define user schema

const userSchema = new mongoose.Schema({

name: { type: String, required: true },

email: { type: String, required: true, unique: true },

password: { type: String, required: true }

}, { timestamps: true });

// Create user model

const User = mongoose.model('User', userSchema);

// Export user model

module.exports = User;

This code defines a Mongoose schema for a user, which includes fields for the user's name, email, and password. It also creates a Mongoose model for the user schema and exports it for use in other parts of your code.

1. Inside the project directory, create a new file called **index.js**, and add the following code to it:

// Require necessary modules

const express = require('express');

const expressLayouts = require('express-ejs-layouts');

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

const User = require('./models/user');

const database = require('./config/database');

// Create Express app

const app = express();

// Set view engine and layout

app.set('view engine', 'ejs');

app.use(expressLayouts);

// Set static folder

app.use(express.static('assets'));

// Use body-parser middleware

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

// Define routes

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

res.render('home', { title: 'Home' });

});

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

res.render('signup', { title: 'Sign Up' });

});

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

const user = new User({

name: req.body.name,

email: req.body.email,

password: req.body.password

});

user.save()

.then(() => {

res.redirect('/');

})

.catch((err) => {

console.error(err);

res.redirect('/signup');

});

});

// Start server

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

app.listen(port, () => {

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

});

This code sets up an Express app, sets the view engine to EJS and the layout to **layout.ejs**, sets a static folder for serving CSS files, and uses the body-parser middleware for processing form data. It also defines routes for the home page and the signup page, as well as a POST route for processing the signup form. The POST route creates a new user object based on the form data, saves it to the database using Mongoose, and then redirects the user to the home page.