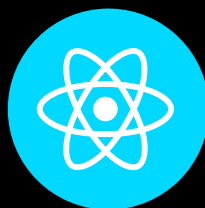
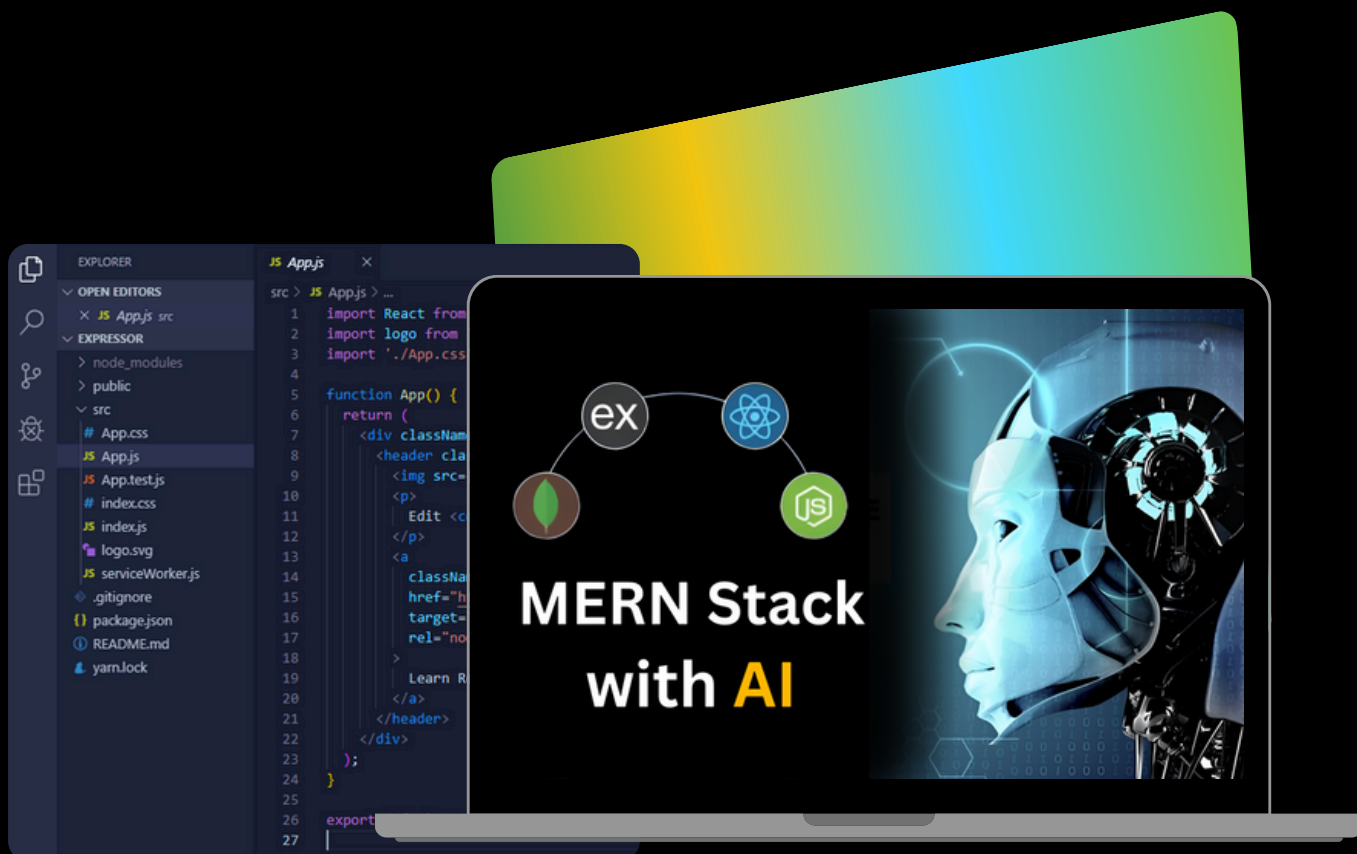


MERN Syllabus

For Apprenticeship Training Program





MERN stands for **MongoDB, Express.js, React.js, and Node.js**. These four technologies come together to form a powerful stack that allows developers to build dynamic, full-stack web applications with ease. Each component in the MERN stack serves a specific role, contributing to a seamless development process that is ideal for building both modern, user-friendly websites and robust, scalable web applications.

MongoDB is a NoSQL database that stores data in a flexible, JSON-like format. It allows developers to handle large amounts of data in a highly scalable and efficient way. Unlike traditional relational databases, MongoDB is schema-less, meaning that data can be easily modified and expanded as your application grows. This flexibility makes it a perfect choice for applications that require the storage and retrieval of unstructured or semi-structured data.

Express.js, a web application framework for Node.js, simplifies the process of building web applications by providing robust features for routing, handling HTTP requests, and managing middleware. Its minimalist approach allows developers to build scalable web applications and APIs quickly, without getting bogged down in the complexities of the server-side code. By using Express.js, developers can focus more on the logic and functionality of their applications rather than worrying about the infrastructure.

React.js, the front-end library in the MERN stack, is renowned for its ability to build fast, interactive, and highly dynamic user interfaces. Developed by Facebook, React allows developers to create reusable UI components that update efficiently as the application's data changes. Its virtual DOM ensures optimal performance, making React a preferred choice for building single-page applications (SPAs) and large-scale web projects that require dynamic, real-time user experiences.

Finally, **Node.js** is a powerful, event-driven JavaScript runtime that allows developers to run JavaScript on the server side. It enables fast and efficient execution of code, which is crucial for handling high volumes of requests in real-time applications. Node.js is built on Chrome's V8 engine, giving it exceptional performance capabilities. Moreover, its non-blocking I/O model makes it ideal for building scalable network applications, where thousands of connections can be handled simultaneously.





HTML : MODULE 1

Introduction:

- What is HTML?
 1. Definition and importance of HTML
 2. Role of HTML in web development
 3. Overview of how HTML works with CSS and JavaScript
- History Of HTML ?
 1. Evolution of HTML from HTML 1.0 to HTML5
 2. Key milestones and developments in HTML history
 3. Impact of HTML advancements on web development
- Basic Structures of a Web Page
 1. Fundamental structure of an HTML document
 2. Understanding the `<!DOCTYPE html>` declaration
 3. Introduction to HTML elements and attributes

HTML Head Tags:

- Purpose and Use of the `<head>` Section
 1. Common head tags: `<title>`, `<meta>`, `<link>`, `<style>`, `<script>`, `<meta viewport="">`
 2. Using meta tags for SEO and responsive design

HTML Body Tags:

- Structure and Purpose of the `<body>` Section
 1. Creating and organising content within the body
 2. Basic HTML tags for structuring web pages

HTML Basic Tags:

- `
` and `<hr>` Tags
 1. Usage and examples of line breaks with `
`
 2. Creating horizontal rules with `<hr>`
- `<p>`, `<h1>` to `<h6>`, `<a>`, ``, ``, ``, ``, `<div>`, ``, ``, ``, `<table>`, `<tr>`, `<td>`, `<th>`, `<form>`, `<input>`, `<button>`

HTML Link Creation:

- HTML Anchor Tags
 1. Creating hyperlinks with the `<a>` tag
 2. Understanding `href` attribute and link destinations
 3. Opening links in new tabs or windows with `target` attribute

HTML Tables

- HTML Tables
 1. Constructing tables with `<table>`, `<tr>`, `<td>`, and `<th>` tags
 2. Basic table structure and usage

```

<!DOCTYPE html>
<html>
  <head>
    <title>Wikitechy div tag</title>
  </head>
  <body>
    <h2>Wikitechy HTML div tag.
    <div style="color: red">
      <h2>This is a heading in
      <p> This is a paragraph i
    </div>
  </body>
</html>
  
```




- HTML Form Tags and Attributes
 1. Essential form tags: `<form>`, `<input>`, `<label>`, `<fieldset>`
 2. Understanding form attributes: `action`, `method`, `name`, `id`
- HTML Forms - Post vs. Get
 1. Differences between `POST` and `GET` methods
 2. Appropriate use cases for each method
- HTML Forms - Input Text Fields
 1. Creating text input fields with `<input type="text">`
 2. Attributes for text fields: `placeholder`, `maxlength`, `required`
- HTML Forms - Select Menus
 1. Creating drop-down lists with `<select>` and `<option>` tags
 2. Handling multi-select menus
- HTML Forms - Checkboxes and Radio Buttons
 1. Using `<input type="checkbox">` and `<input type="radio">`
 2. Grouping radio buttons for exclusive selection
- HTML Forms - Text Areas and Buttons
 1. Creating multi-line text input with `<textarea>`
 2. Adding buttons: `<button>`, `<input type="submit">`, `<input type="reset">`

TAILWIND CSS FUNDAMENTALS : MODULE 1

Introduction to Tailwind

- What is Tailwind CSS?
- Why use utility-first CSS?
- Comparison: Tailwind vs. traditional CSS

Setting Up Tailwind

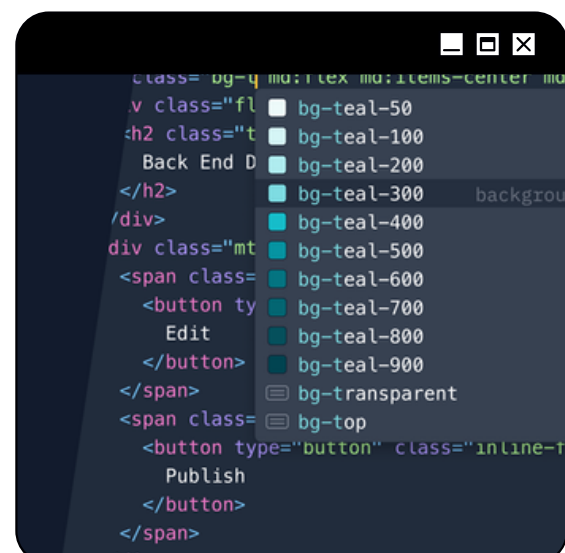
- Create a basic HTML file
- Add Tailwind CDN link
- Verify Tailwind classes work (e.g., `bg-red-500`)

Working with Utility Classes

- Text & Font: `text-lg`, `text-red-500`, `font-bold`
- Background & Color: `bg-blue-500`, `text-white`, `bg-gray-100`
- Spacing: `m-4`, `p-2`, `mt-8`

Border and Radius Utilities

- Add borders: `border`, `border-2`
- Customize radius: `rounded`, `rounded-full`
- Border color & shade: `border-gray-300`, `border-red-500`





TAILWIND CSS LAYOUT & POSITIONING : MODULE 2

Flexbox Layout

- Enable flex: flex
- Center content: justify-center, items-center
- Direction and gap: flex-col, gap-4

Grid Layout

- Basic grid: grid, grid-cols-2
- Responsive grids: md:grid-cols-3, lg:grid-cols-4
- Grid spacing: gap-2, gap-x-4, gap-y-6

Responsive Design

- Screen prefixes: sm:, md:, lg:
- Example use: text-sm md:text-lg
- Hide/show: hidden md:block

Position and Alignment

- Use relative, absolute
- Align content with text-center, text-right
- Center items vertically using flexbo

TAILWIND CSS INTERACTIONS & PROJECT : MODULE 3

Hover and Focus Effects

- Background hover: hover:bg-green-500
- Text hover: hover:text-blue-700
- Focus styles: focus:outline-none, focus:ring

Shadow & Transitions

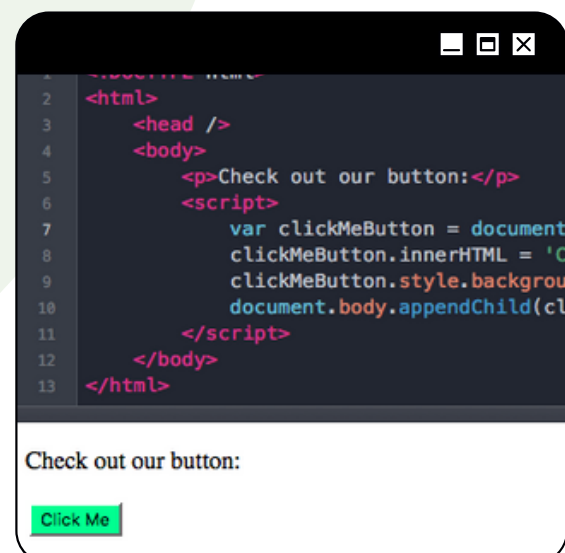
- Basic shadow: shadow, shadow-md
- On hover: hover:shadow-lg, hover:shadow-xl
- Smooth transition: transition, duration-300, ease-in-out

Button and Card Styling

- Buttons: bg-blue-500 text-white px-4 py-2
- Cards: p-6 shadow rounded-lg
- Hoverable cards: hover:bg-gray-100

Mini Project – Landing Page or Card

- Use layout utilities (flex/grid)
- Make it responsive with sm:, md:, lg:
- Add hover effects, spacing, and transitions





JAVASCRIPT : MODULE 1

Introduction: In this module we will learn about what JavaScript is and the benefits of the language. We can also learn how to write the first JavaScript program.

- JavaScript Introduction
- First Javascript program
- Benefits of Javascript

JAVASCRIPT : MODULE 2

Language Syntax: In this module we will learn about JavaScript language syntax and how to use the program level. And we can also learn to understand the arrays.

- Variable declaration
- Operators
- Control Statements
- Error Handling
- Understanding arrays
- Function Declaration

JAVASCRIPT : MODULE 3

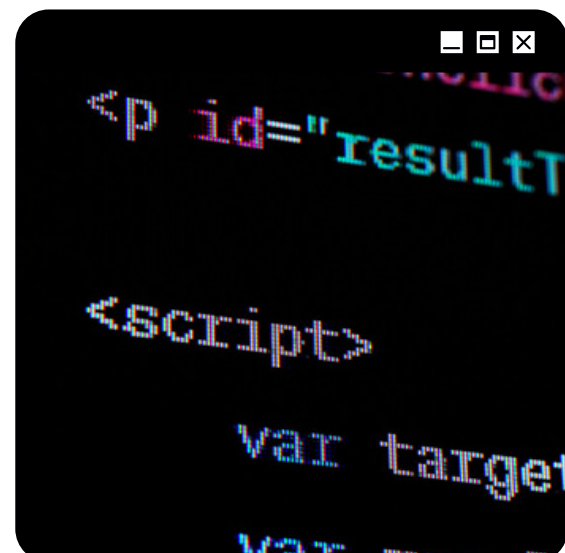
HTML Forms: In this module we will learn about how to use the forms. What are the properties in JavaScript? HTML form validation can be done by JavaScript.

- HTML Document object Model
- Working with HTML form and its elements

JAVASCRIPT : MODULE 4

HTML DOM: In this module we will learn about how JavaScript can access and change all the elements of an HTML document.

- HTML Document object Model
- Working with HTML form and its elements





INTRODUCTION : MODULE 1

Introduction

What is the "MERN STACK"?

MERN - A FIRST OVERVIEW

Course Outline

How To Get The Most Out Of The Course

Using the Course Resources

Course Setup

THE MERN STACK : MODULE 2

Module Introduction

Understanding the Big Picture

Diving Into the Frontend

Understanding the Backend

REST vs GraphQL

Connecting Node & React

MERN - Essentials

Creating our Development Environment & the Development Servers

Diving Deeper Into the Code

Useful Resources & Links

PLANNING THE APP : MODULE 3

Module Introduction

Understanding the General App Idea

Sketching out the Frontend

Data & API Endpoints used in our App

Required SPA Pages for the Frontend

REACT.JS - A REFRESHER : MODULE 4

Module Introduction

What is React?

A Note About The NodeJS Version

React 18

Setting Up a Starting Project

Understanding JSX

Understanding Components

Working with Multiple Components

Using Props to pass Data between Components

Rendering Lists of Data

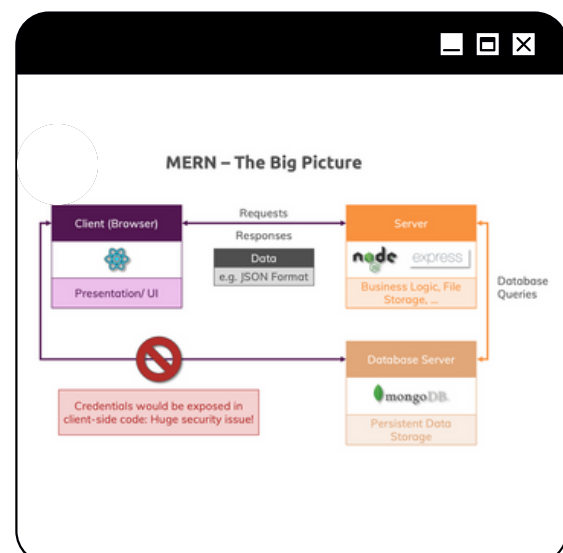
Handling Events

Efficient Child<=>Parent Communication

Working with "State"

More on State

Fetching User Input (Two-way Binding)





React.js Basics

Useful Resources & Links

REACT.JS - BUILDING THE FRONTEND : MODULE 5

Module Introduction

Important: Use React Script v5

Starting Setup, Pages & Routes

Adding a UsersList Page / Component

Adding a UserItem Component

Styling our App & More Components

Presentational vs Stateful Components

Adding a Main Header

Adding Navlinks

Implementing a Basic Desktop & Mobile Navigation

Understanding Portals

Handling the Drawer State

Animating the Sidedrawer

Rendering User Places & Using Dynamic Route Segments

Getting Route Params

Adding Custom Buttons

Adding a Modal

Rendering a Map with Open Streets Maps

Continuing without a Credit Card

Optional: More on the useEffect() Hook

Adding a Custom Form Input Component

Managing State in the Input Component

Adding Input Validation

Sharing Input Values & Adding Multiple Inputs

Managing Form-wide State

Finishing the "Add Place" Form

Starting Work on the "Update Place" Page

Adjusting the Input Component

Creating a Custom Form Hook (useForm)

Optional: More on (Custom) React Hooks

Adjusting the Form Hook

Fixing Minor Issues

Showing a Deletion Warning

Adding an "Auth" Page & Login Form

Adding Signup + "Switch Mode" Button

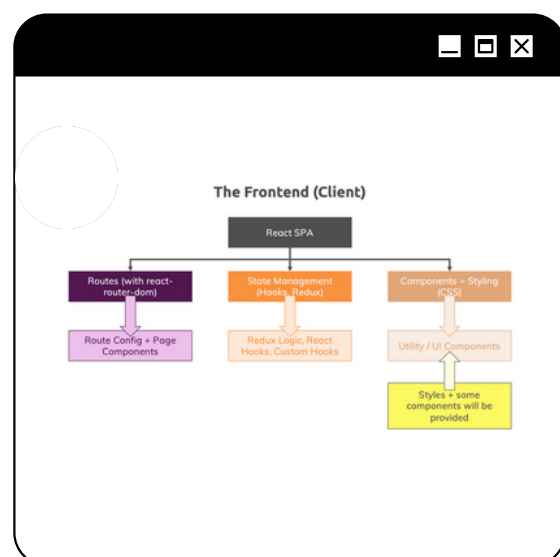
Adding Auth Context for App-wide State Management

Listening to Context Changes

Adding Authenticated & Unauthenticated Routes

More Auth Context Usage

Useful Resources & Links





NODE.JS & EXPRESS.JS - A REFRESHER : MODULE 6

Module Introduction

What is Node.js?

Writing our First Node.js Code

Sending Requests & Responses

What is Express.js?

Adding Express.js

Understanding the Advantages of Express.js

How Code Execution Works

Node & Express Basics

Useful Resources & Links

NODE.JS & EXPRESS.JS - BUILDING OUR REST API : MODULE 7

Module Introduction

Setting up our Project

Implementing Basic Routing

Adding Place-Specific Routes

Getting a Place By User ID

Handling Errors

Adding our own Error Model

Adding Controllers

Adding a POST Route & Using Postman

Handling Errors for Unsupported Routes

Adding Patch Routes to Update Places

Deleting Places

Finalizing the "Get Places by User ID" Resource

Setting Up the User Routes (Signup, Login, Get Users)

Validating API Input (Request Bodies)

Validating Patch Requests & User Routes

Useful Resources & Links

WORKING DEEP WITH MONGO DB & MONGOOSE : MODULE 8

Module Introduction

What is MongoDB?

SQL vs NoSQL

Connecting React to a Database?

Setting Up MongoDB

Creating a Simple Backend & with the Database

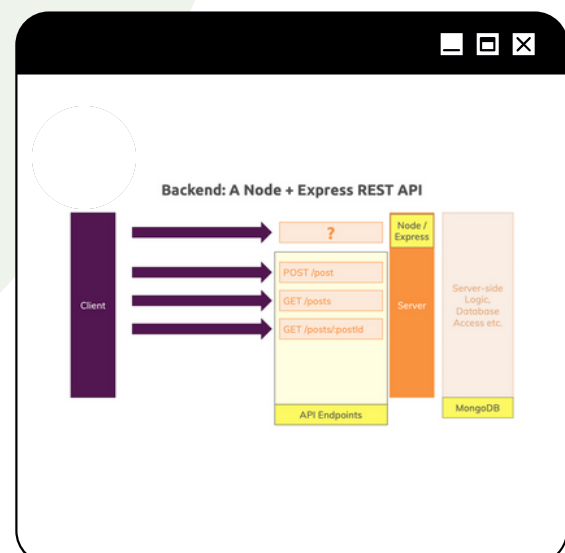
Creating a Document with MongoDB

Getting Data from the Database

Installing Mongoose

Understanding Models & Schemas

Creating a Product





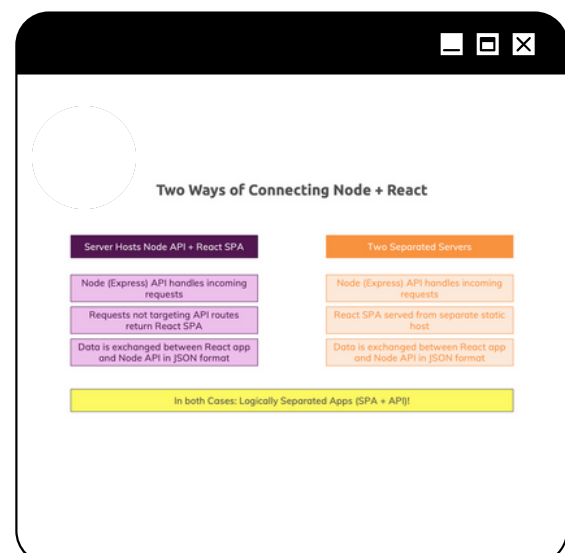
Connecting to the Database & Saving the Product
Getting Products
Understanding the ObjectID

CONNECTING THE BACKEND TO THE DATABASE - MONGO DB & MONGOOSE : MODULE 9

Module Introduction
Installing Mongoose & Connecting our Backend to MongoDB
Creating the Place Schema & Model
Creating & Storing Documents in the Database
Getting Places by the PlaceID
Getting Places by the UserID
Updating Places
Deleting Places
How Users & Places are Related
Creating the User Model
Using the User Model for Signup
Adding the User Login
Getting Users
Adding the Relation between Places & Users
Creating Places & Adding it to a User
Deleting Places & Removing it From the User
Getting Places - An Alternative
Cleaning Up our Code
Useful Resources & Links

CONNECTING THE REACT.JS FRONTEND TO THE BACKEND : MODULE 10

Module Introduction
Initial Setup
Sending a POST Request to the Backend
Optional: The fetch() API
Handling CORS Errors
Getting Started with Error Handling
Proper Error Handling in the Frontend
Sending a Login Request
Getting Users with a GET Request
Creating a Custom Http Hook
Improving the Custom Http Hook
Using the Http Hook to GET Users
Adding Places (POST)
Loading Places by User Id
Updating Places
Deleting Places
Fixing NavLinks & "My Places"
Final Adjustments
Useful Resources & Links





WHAT IS AN AI IMAGE GENERATOR?

An AI image generator uses machine learning models to create images based on textual descriptions or other input data.

OVERVIEW OF THE MERN STACK FOR THE PROJECT

The MERN stack consists of MongoDB (database), Express (backend framework), React (frontend framework), and Node.js (runtime environment).

SETTING UP THE FOLDER STRUCTURE FOR A MERN PROJECT

The folder structure includes separate directories for backend (server), frontend (client), and shared components like utilities or assets.

INSTALLING NECESSARY DEPENDENCIES FOR THE BACKEND AND FRONTEND

Use npm to install dependencies like Express for the backend, and React along with MUI for the frontend.

CREATING A GIT REPOSITORY FOR VERSION CONTROL

Initialize a Git repository and commit your project to maintain version control and collaborate efficiently.

BACKEND DEVELOPMENT (NODE.JS + EXPRESS + MONGODB)

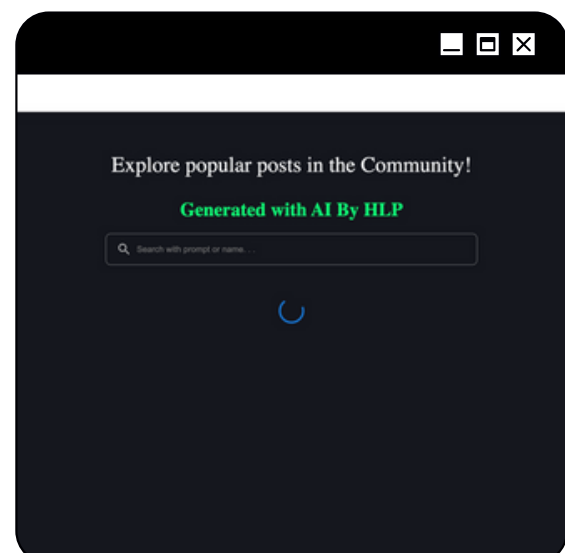
Set up Node.js and Express to create the backend API that communicates with MongoDB to store and retrieve data.

SETTING UP UNSPLASH API, CLOUDINARY

Obtain credentials for Unsplash API and Cloudinary, and securely store MongoDB credentials to access the services.

CREATING API ENDPOINTS FOR FETCHING AND STORING IMAGES

Implement API endpoints to interact with the image sources (Unsplash and Cloudinary), allowing image fetch and storage operations.





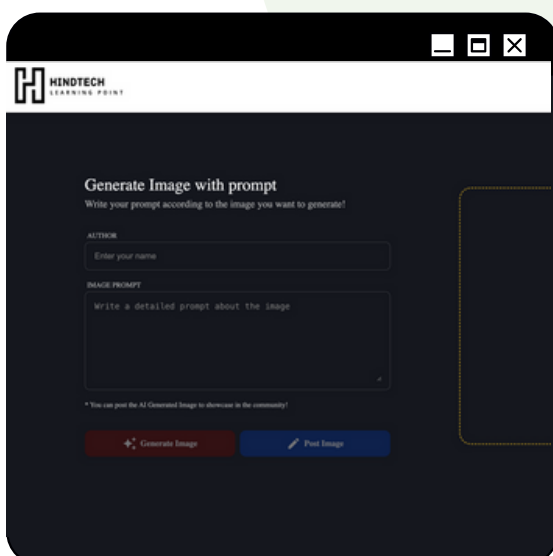
PROJECT DEVELOPMENT

1. **AI Chat & Gen (MAJOR PROJECT 1)**
2. **YOUR PLACES (MAJOR PROJECT 2)**

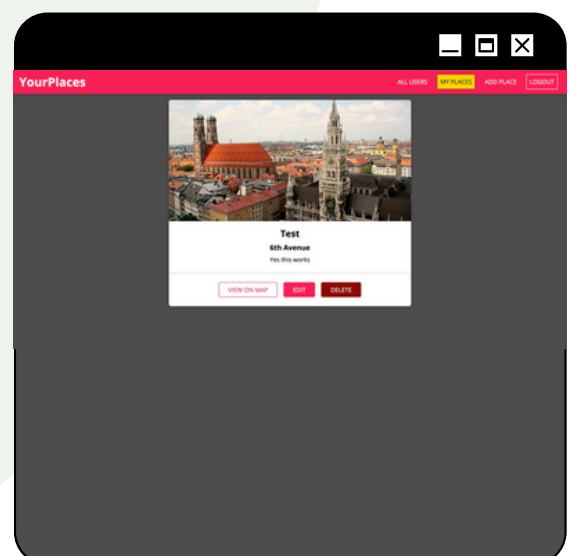
DURATION : 180 DAYS

OUTCOME:

- **Build Fullstack MERN Apps:** Master the MERN stack (MongoDB, Express, React, Node) to create powerful fullstack applications from scratch.
- **Add Authentication & File Uploads:** Learn how to implement user authentication, authorization, and file uploads for real-world web applications.
- **Deploy Fullstack Projects:** Understand various deployment strategies to take your MERN stack applications live on different hosting platforms.
- **Master Database Integration with MongoDB:** Understand how to effectively integrate MongoDB with Node.js and Express to manage data seamlessly in fullstack web applications
- **Real-Time Features with Socket.io:** Learn how to integrate real-time functionalities such as chat systems.



MAJOR PROJECT 1



MAJOR PROJECT 2

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.



Hindtech Learning Point

A Trusted IT Institute Brand In Lucknow

Admission Open For

Summer Training

CSE/IT

Diploma, B.Tech/B.E, BCA, MCA

45 - 50 Days Program

Apprenticeship Training

CSE/IT

B.Tech, BCA, MCA, PGDCA, Dip.

6 to 8 Months Program

Industrial Training

CSE/IT

B.Tech, BCA, MCA, PGDCA, Dip.

3 to 5 Months Program

UI/UX Designing

CSE/IT or interested student

B.Tech, BCA, MCA, PGDCA, Dip.

45 - 50 Days Program

1
Number
ONE

Excellent Performance
by Hindtech Learning Point
in IT Training



Since 2020, 1000+ students placed in top IT firms | Parmar Plaza, BKT, Bargadi, UP 226201 | Call: 6307738600 / 7905320279 | www.hindtechlearningpoint.com



HINDTECH

IT SOLUTIONS

Training

MERN Apprenticeship Training Program

Duration

6 to 8 Months

Amount

₹21,000



www.hindtechitsolutions.com



hr@hindtechitsolutions.com



+91 7905320279



www.hindtechlearningpoint.com



hindtechlearningpointlko@gmail.com



+91 6307738600



Building No 10/703, Ground Floor, near Arvindo Park Road, Sector 10, Indira Nagar, Lucknow, Uttar Pradesh 226016

Branch 1



PARMAR PLAZA, 1st Floor, Bakshi Ka Talab, Bargadi Magath, Uttar Pradesh 226201

Branch 2

Scan Me:



Scan this QR code for details



HINDTECH
LEARNING POINT

