



Full Stack Software Development

MERN

The MERN stack is a JavaScript-based framework for developing web applications. MERN is named after MongoDB, Express, React, and Node, the four key technologies that make up the layers of the stack.





ABOUT

Acquiring Technology was developed and is the world's first cloud-based Ed-Tech learning platform, allowing students to master programming skills in their native language. In order to democratize tech education worldwide, it meets the EdTech industry's standards.

Our Mission

To democratize tech education worldwide by meeting the standards of the EdTech industry.

Our Vision

To shape lives by bestowing high-end tech skills to learners in their native languages & Connect the tech career aspirants with the corporate industry.

ABOUT TRAINER

Our Live Class is packed with Capstone Project Based on Full Stack Development Program that offers 100% Job Assistance Support on completing the course. In addition to mentoring from industry experts from top MNC companies, the program has been designed by Founders (former Disney developers). While it will assist you in getting hired by top MNC's. Enhance your career in technology.



MODULE 1

HTML, CSS AND Bootstrap CSS From Level 0

Build a Static Version of Acquiring technology Like Website

Interview Preparation Module 1 (HTML + CSS + Bootstrap CSS)

Mini Project



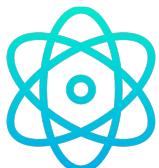
MODULE 2

JavaScript

Making The Website Dynamic Using JavaScript

Interview Preparation Module 2 (JavaScript)

Mini Project



MODULE 3

ReactJS

Redux

Deployment

Interview Questions (ReactJS)

Mini Project



MODULE 4

NodeJS

ExpressJS

MongoDB

Building Website Using NodeJS, ExpressJS

Make Database For a Website Using MongoDB

Interview Preparation (Nodejs, Express js, MongoDB)

Mini Project



PROJECT 1

How to create a portfolio using Html CSS



PROJECT 2

How to create a Reestar Restaurant project using Html CSS



PROJECT 3

How to Create To-Do List for Your Daily Activities



PROJECT 4

Nostra E-Commerce Project Sale your Own Products



PROJECT 5

Create a login and signup functionality for a React-based Actodo project.



PROJECT 6

Develop a weather application using React and the Axios API.



PROJECT 7

A bulk mail application utilizing Nodemailer, React, Node.js, MongoDB, and Express



PROJECT 8

Netflix Clone using Html CSS and Bootstrap CSS Responsive Design



PROJECT 9

Bookmyshow Creating a Stylish Web Interface with HTML, CSS, and Bootstrap CSS

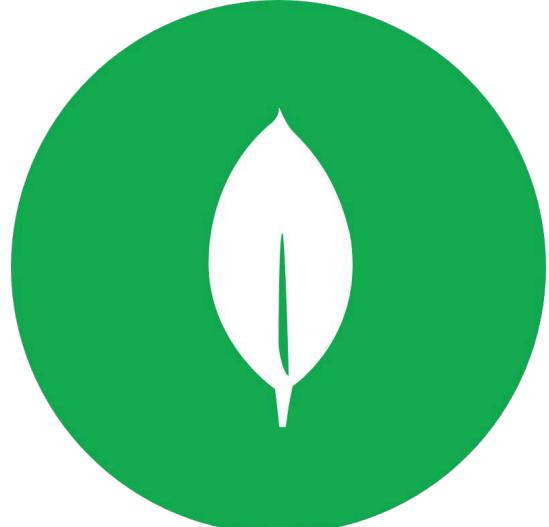
Final Project Task

amazon like E-commerce



- 1. Secure Authentication:** User JWT ensures safe logins.
- 2. Robust Backend:** MongoDB for efficient data handling.
- 3. Admin Privileges:** Add products easily.
- 3. Efficient Checkout:** Razorpay integration for smooth payments.
- 4. Efficient Checkout:** Razorpay integration for smooth payments.
- 5. Dummy Razor API:** Test transactions worry-free

Join us for a top-notch shopping journey!



Unlock Your Potential with Our Internship Opportunity

Upon successfully completing all assigned projects, students who stand out in terms of performance will be eligible for a one-month internship, complete with a stipend. This opportunity not only allows you to apply what you've learned in a real-world setting but also emphasizes the importance of timely project completion as a criterion for selection.

Completion of Projects: You must successfully complete all given projects within the specified deadlines to qualify for the internship opportunity

Internship Duration and Stipend: The internship lasts for one month. It's a paid opportunity, allowing you to earn while you learn and apply your skills

Selection for Internship: Selection is based on your performance in the course and the projects. It's your chance to shine and show us what you've got!

Job Placement Assistance: Post-internship, based on your performance during the interview process, we will offer assistance with job placements, leveraging our network to help you kickstart your career in the IT industry

Exclusive Community Access: Interns will gain access to a separate community, creating a space for networking, collaboration, and shared learning among peers and mentors.



Section 1: Let's Dive into HTML

1.2 HTML Introductio

1.2 HTML in Action - Learning Basic tags

1.3. Building a simple portfolio website using only HTML

1.4 Browser Developer Tools

1.5. HTML Boilerplate

Section 2: HTML Forms and Tables

2.1 HTML Forms

2.2. HTML Tables

2.3 HTML Tables + Forms

2.4. Create a Contact form for my Portfolio Page

2.5. How to link Multiple HTML Pages?

2.6. Things you should kn





Section 3: Introduction to CSS

- 3.1. What is CSS and its type?
- 3.2. CSS Basic Styles
- 3.3. CSS Selectors
- 3.4. CSS Box Model Explained
- 3.5. Display Property of CSS is everything you need
- 3.6. CSS hover Property
- 3.7. Creating a CSS Card Component
- 3.8. BEM Explained
- 3.9. The concept of Specificity



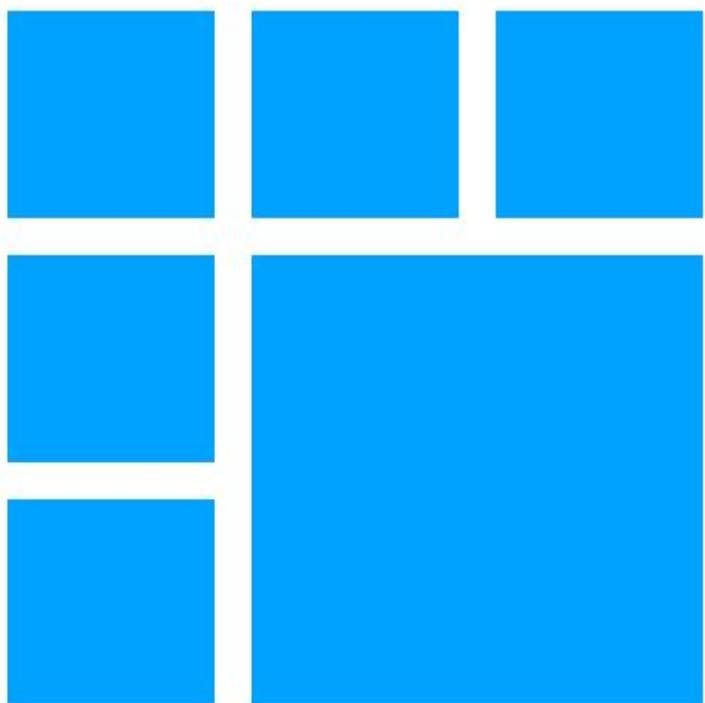


Section 4: Upgrading Portfolio Website

- 4.1. Adding basic Styles
- 4.2. Working with Display Property
- 4.3. Styling Contact Page
- 4.4. Launch your first website on GitHub
- 4.5. Restart Project Task

Section 5: Master Flexbox with a Project

- 5.1. Udemy Project Introduction
- 5.2. Introduction to Complete Flexbox with Exercise
- 5.3. Building Categories Section - Netflix
- 5.4. Sale Image Section - Netflix
- 5.5. Course Card - Netflix
- 5.6. Flex-grow and Flex-basis Explained
- 5.7. Wrapping up Netflix Project
- 5.8. Time to Practice Flex





Section 6: CSS Positioning with Examples

- 7.1. What is CSS Position
- 7.2. What are 5 Different Positions?
- 7.3. Create Components Using CSS Position
- 7.4. What is z-index?
- 7.5. Update Netflix Project and Git it.

Section 7: Git and GitHub - Introduction

- 7.1. What is Git and GitHub?
- 7.2. Setup your Git Environment
- 7.3. GitHub Push and Pull Demo
- 7.4. Let's Push Netflix Project from the Terminal



PROGRAM

CSS Animations & Media Queries



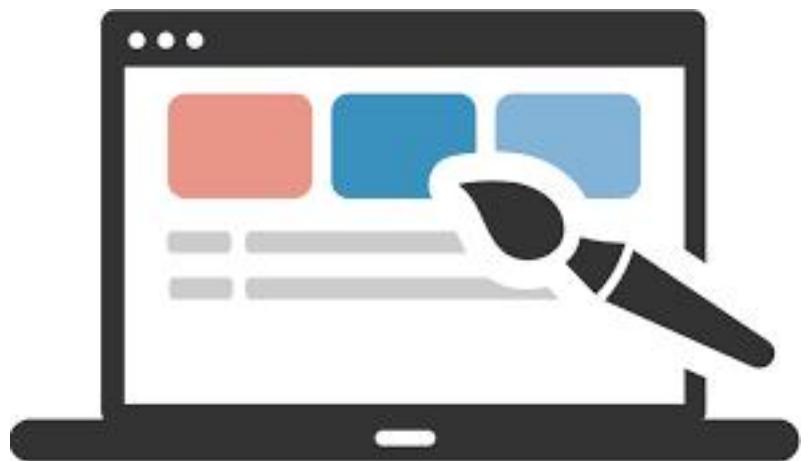
ACQUIRING
TECHNOLOGY

Section 8: CSS Animations - Learn Basics

- 8.1. What is CSS Animation
- 8.2. Opacity vs rgba
- 8.3. Transition Property
- 8.4. Advance Animation using Keyframes.
- 8.5 Animation in Practice.
- 8.6. Update Netflix Project and Git it

Section 9: Responsive Websites Makes You Pro

- 10.1. What is a Responsive Website?
- 10.2. Introduction to Media Queries
- 10.3. Practice Questions on Responsive Design
- 10.4. Upgrading Udemy Project using MQ



Section 10: Time to CSS all by yourself

10.1. You are going to create a TripAdvisor Clone

Section 11: Bootstrap for Fast Development

11.1. What is Bootstrap CSS?

11.2. Introduction of Grid & Columns

11.3. Introduction to E-Commerce Project

11.4. Building Responsive Navbar

11.5. Header Section

11.6. Dynamic JavaScript Components

11.7. Product Page Task

11.8 How you can easily Take Template from Internet

11.9 Time to Git it



Section 12: JavaScript: It's time to code.

- 12.1. Why JavaScript?
- 12.2. What are Variables?
- 12.3. What ES6 means?
- 12.4. JavaScript Primitives
- 12.5. JavaScript Operators
- 12.6. Ternary Operators
- 12.7. Let's Practice Basic JavaScript
- 12.8 How you can easily Take Template from Internet
- 12.9 Time to Git it

Section 13: JavaScript: If-Else and Functions

- 13.1 If-Else in JavaScript
- 13.2 If-Else Practice Questions
- 13.3 What are functions in JavaScript?
- 13.4 Parameters Explained
- 13.5 Return Keyword in JavaScript
- 13.6 Function Practice Questions

Section 14: JavaScript: Guess the Number Game

- 14.1. How to generate a random Number?
- 14.2. Let's create a Game





Section 15: JavaScript: Loop is powerful.

15.1. Loops Explained!

15.2. Exercise Questions.

Section 16: JavaScript: Arrays

16.1. What is Array?

16.2. Push and Pop

16.3. Slice and Splice

16.4. indexOf, includes, Length.

16.5. const & Arrays

16.6. Combining Arrays with loop

16.7. Arrays Practice Question

Section 17: JavaScript: Object Literals

17.1. Introduction to Object Literals

17.2. Arrays + Objects



Section 18: JavaScript: DOM Introduction

18.1. What is JavaScript DOM

18.2. Selecting and Manipulating DOM using ID

19.3. Selecting and Manipulating DOM using querySelector

19.4. textContent vs InnerHTML

19.5. Exercise Questions with Id and querySelector

19.6. use of setAttribute?

19.7. How to Manipulate CSS?

Section 19: All about Events

19.1 What are Events & Even handlers?

19.2. Event Object

19.3. Add 2 Numbers and Print Result

19.4. Exercise with Input Tag

19.5. Guess the Random Number 2.0

19.6. What are event Listeners with Example?

19.7. Create a PopUp Box on a Button Click - 1

19.8. Create a PopUp Box on a Button Click - 2

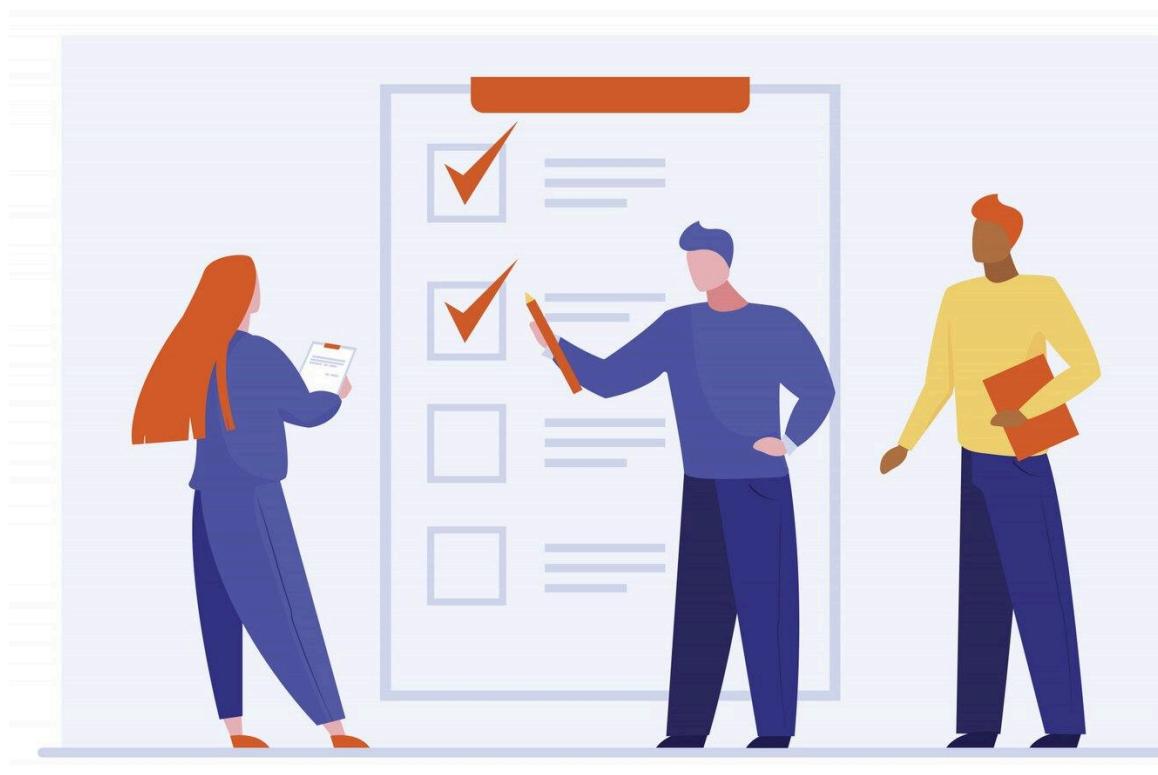


Section 20: Let's Create a TO-DO List

- 20.1. How to create a HTML Element?
- 20.2. Append vs Prepend.
- 20.3. How to Delete an Element
- 20.4. Let's Create a TO-DO List from Scratch
- 20.5. Task for the Session

Section 21: Working with Forms

- 21.1. How to select value from Radio Buttons
- 21.2. How to select value from CheckBox
- 21.3. What is Regex and its Example
- 21.4 Form Validation



Section 22: Upgrading Greenden Project with JS

22.1 Creating a Responsive Side Navbar

22.2. Search Functionality in Product Page

22.3. Deploy into GitHub

Section 23: E-commerce Task (HTML+CSS+JS)

23.1. Create an E-commerce Website called Nostra

Section 24: ES6 in detail

24.1. Arrow Function

24.2. Template Literals

24.3. Destructuring Assignment

24.4. Spread and Rest Operator

24.5. Default Parameter

24.6. Classes

24.7. Callback

24.8. Promises

24.9. Async /Await



Section 25: JavaScript Array Methods

25.1. ForEach Method

25.2. Let's Filter

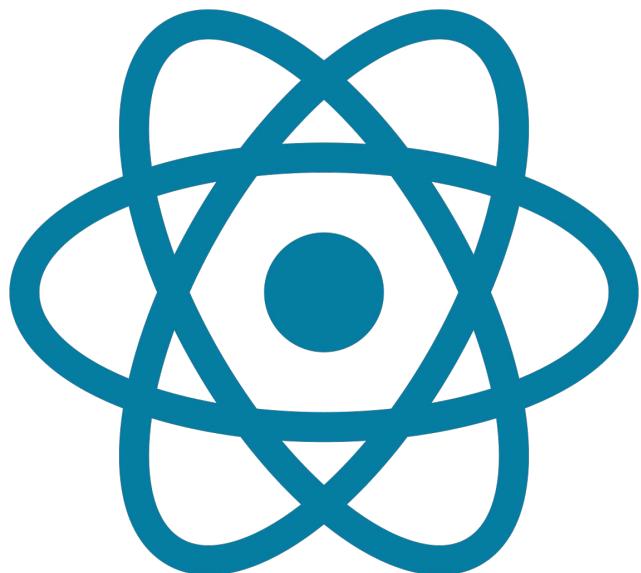
25.3. Map is powerful

Section 26: This is what everyone is talking about - React JS

- 26.1. What and why is React?
- 26.2. Let's Practice React - codesandbox
- 26.3. How to run React Application from VS Code
- 26.4. How to add Styles
- 26.5. What are React Components?
- 26.6. Recreate Perfumy Project Using React
- 26.7. Import and Export Components.
- 26.8. Upgrade Perfumy React
- 26.9. What are Fragments?
- 26.10. Push React Apps to Git
- 26.11. Using vercel to Deploy
- 26.12. Convert Udemy css Project to Netflix-react

Section 27: Props and Map

- 28.1. What are props?
- 28.2. Separation of Components
- 28.3. Statement vs Expression
- 28.4 How to Work with Maps





Section 28: Exercise Question: React Components, Props & Map

29.1. Image Gallery Exercise.

Section 29: useState Explained.

30.1. What is state

30.2. How to create a State

30.3. Counter App with State

30.4. State Exercise



Section 30: Form Controls

31.1. Controlled Components

31.2. Multiple Inputs

31.3. Multiple Inputs 2.0

31.4. Creating a Shopping List

Section 31: Conditional Rendering

31.1. Change Color on a Button Click

31.2. Check & Uncheck List

31.3. Before vs After Login

Section 33: Things you should know

33.1. React component Life Cycle

33.2. React Hooks and its Types.

33.3. UseEffects Hook Explained

33.4. List and key

Section 34: Actodo Project - Part 1

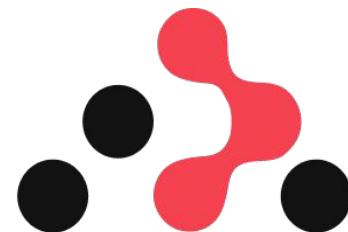
- 35.1. Actodo Project Overview
- 35.2. How to Integrate Bootstrap with React
- 35.3. Project Folder Structure
- 35.4. Creating a Header Component
- 35.5. Creating a Card Component
- 35.6. Building TodoList
- 35.7. Adding TodoList Functionality
- 35.8. Building AddTodoForm

Section 35: React Router

- 35.1 React router Explained.
- 35.2. Link Tag in React Router
- 35.3. Create Login and Signup Page
- 35.4. Add Login Functionality
- 35.5. Add User via Sign Up
- 35.6. Fixing Login Logical Error
- 35.7. How to Navigate?

Section 36: Redux Tutorial

- 36.1. What and why is Redux?
- 36.2. Redux Short Tutorial
- 36.3. Redux Slicer Explained
- 36.4. Redux Mini Project



React Router

PROGRAM

React Router & Api & Node Js

Section 37: Finishing up Actodo Project

37.1 Merge Router+Actodo Todo List

37.2 Let's Deploy

Section 38: useContext Hook Explained

38.1. Why useContext

38.2. Create your own context

38.3. Add list with UseContext.

Section 39: Task (Router + Context)

39.1 Favourite Student List Task

Section 40: Weather App using API

40.1. What is API?

40.2. What are Promises?

40.3. How to work with Axios?

40.4. Working with Weather API

40.5. React Weather App

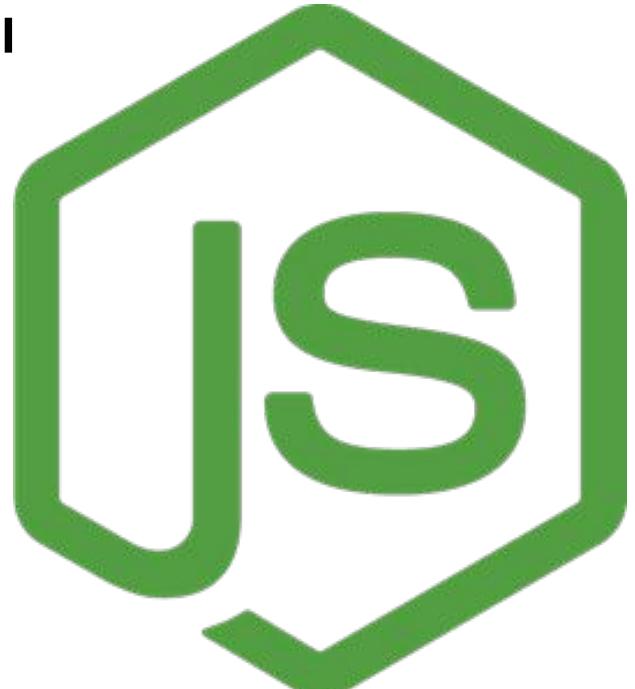
Section 41: Backend Begins

41.1. Server vs Client

41.2. How to create your own server

41.3. Let's Create an API

41.4. Practice Session



Section 42: Integrate Backend and Frontend

42.1. Let's Connect Backend with html

42.2. What is Get and POST Method?

42.3. Simple Login Practice.

42.4. How to receive data from Post?

Section 43: Integrate React with Node

43.1. Create Login Functionality in React

43.2. What is Middleware?

43.3. Send and receive data from React and Node

43.4. Axios Post Method

Section 44: React + Node Exercise

44.1. Recreate Todo List with React + Node.

Section 45: Let's Learn about Database

45.1. What is Database?

45.2. Relational vs non-relational



Section 46: MongoDB

- 46.1. What and why is MongoDB
- 46.2. Installation and Setup
- 46.3. MongoDB CRUD
- 46.4. MongoDB CRUD Exercise

Section 47: Integrate MongoDB with Node

- 47.1. How to use Mongoose?
- 47.2. Creating Todo with React + Node.
- 47.3. Connecting Todo with MongoDB

Section 48: Creating BulkMail App (MERN APP)

- 48.1. BulkMail App Overview
- 48.2. How to Send an Email from node?
- 48.3. How to read a file?
- 48.4. Building UI for BulkMail
- 48.5. Merge Frontend with Backend
- 48.6. Adding Bulk Mailing Feature
- 48.7. Fixing a Problem
- 48.8. Connecting to MongoDB
- 48.9. MongoDB Cloud
- 48.10 How to Deploy?



PROGRAM

Interview Prep



ACQUIRING
TECHNOLOGY

Section 49: Bonus Session

49.1. Effective way to Build Resume

49.2. Proven Pathway for Job Search

49.3. Building LinkedIn

49.4. Optimizing your profile on platforms like Naukri

49.5. Different Platforms where you can apply and search for a Job

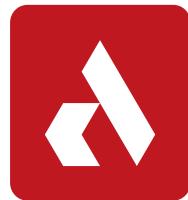
49.6. HR Interview Question and Answer Preparation

49.7. Mock Interviews

49.8. Technical Interview Questions

49.9. Opportunity to attend Direct Interview with our Partner Companies





**ACQUIRING
TECHNOLOGY**

For further details, **contact**



9042771660



acquiringtechnology@gmail.com

www.acquiring.in