UNLIMITED POWERFUL LEARNING

UPL PROSPECTOUS

early break even | best experience

2023

@unlimited powerfull learning

ABOUT COMPANY:

UNLIMITED POWER FULL LEARNING (UPL) aims to solve the challenges and minimize the gap between students with IT industries' expectations. This organization is built by a strong team who are having good academic and industry experience of more than two decades. The founder of this experience G.D. Mallikarjuna has 20+ plus started as a technologist having diverse experience in the education sector as Trainer and Developer.

VISION:

At UPL@SNIPE, we make the best experience in technology learning with career guidance for their life journey

MISSION:

Learn with Live experience and career values.

PROGRAMS OFFERED:

PROGRAMS	DURATION	AMOUNT + GST
CODING BOOT CAMP	4 TO 6 MONTHS	Rs.30000/-
CERTIFICATION COURSE	3 SEMESTERS 1 YEAR COURSE	Rs. 25000/- per semester Rs. 10000/- final semester
CAREER BRIDGE	3 MONTHS	Rs. 50000/-
INDUSTRY READINESS PROGRAM	3 MONTHS	Rs.20000/-

*Terms and Conditions are subjected to further discussion before signing the legal agreement

UPL
UNLIMITED POWER FULL LEARNING

CODING BOOT CAMP

ABOUT THIS MODEL

- **Category:** Virtual Program
- **Target Audience:** Fresher & Experienced
- **Duration:** 4 To 6 Months
- Cost: Rs. 30,000/Candidate (**Registration:** 10K + GST **After 6 Weeks:** 10K + GST **Live Project:** 10K + GST)
- **Course Coverage:** 2 Months training in a relevant discipline, 1 capstone project & followed by involving in live project for duration 4 months.
- **Outcome:** Build their careers feature strong growth projections & lucrative salaries
- **Career Opportunities:** The best jobs you can secure after completing one of these programs such as, Technical Support Specialist, Digital Marketer, Junior Developer, Data Analyst, Web Developer, Project Manager, User Interface/Xxperience (UI/UX) designer, Application Developer, Product Manager, Software Engineer, Full Stack Developer, Data Scientist, Development Operations (DevOps) Engineer, Back End Eeveloper, Teach Others, also Freelancer

COURSES ARE :

- JAVA FULLSTACK
- FULL STACK C# .NET
- FRONT END DEVELOPER IN (REACT/ANGULAR)
- MEAN STACK
- PYTHON
- DATA-SCIENCE
- AUTOMATION TESTING WITH JAVA
- UI/UX DESIGN
- DIGITAL MARKETING
- JENKINS
- MACHINE LEARNING
- DATASTRUCTURE IN PYTHON
- TABLEOU
- POWER BI
- PSPARK
- DEVOPS

*Terms and Conditions are subjected to further discussion before signing the legal agreement

UPL
UNLIMITED POWER FULL LEARNING

BENEFITS IN THIS PROGRAM :

- Uplsnipe Coding Bootcamp Certificate.
- Program Transcript For The Entire Learning Path.
- Coding Bootcamps Can Open Doors To Exciting Technical Career Opportunities.
- Mastering Programming Languages And Associated Technologies Can Prepare You To Work As A Software Or Web Developer.
- Strong Growth Projections And Lucrative Salaries

FULL-STACK DEVELOPMENT WITH REACT / ANGULAR WITH NODE JS

FRONT END DEVELOPER IN (REACT/ANGULAR)

UNIT_001 : 03 HRS

(HTML, CSS, BHOOTSRAP, REACT/ANGULAR) The project, Domain, Platform,Product, Technology, Tech Stack, Process, Frontend Introduction

UNIT_002 : HTML 03 HRS

INTRODUCTION,History,Text-editor,Webbrowser, Html page structure, Html document format, Basic tags list, Html elements and tags,, Types of tags, Html attributes

UNIT_003 : CSS 03 HRS

Introduction, Selector, Style, Background, Border, Display, CSS Float, Font, Text, Button, Table, Form

UNIT_004 : BOOTSTRAP 06 HRS

History,Why bootstrap,Getting started,containers,Grid basic,Typography Colours,Tables,Images,Alerts,Buttons,Progressbar,Spinner,Pagination,Listgroups,Cards,Dropdowns,Collapse,Navs,Carousel,Modal,Tooltip,Popovers,Toast,Scrollspy,Offcanvas,Utilities,Bootstrap 5 forms

*Terms and Conditions are subjected to further discussion before signing the legal agreement

UPL
UNLIMITED POWER FULL LEARNING

UNIT_005 : JAVA SCRIPT — 05 HRS

(Introduction,History,Features,Browser,Browser Object Model (BOM), How does Java script works?,Document Object Model (DOM) ,Data type, undefined, falsy values, conditional and control statements,with statement,looping constructs,string, Arrays,window, forms, events,DHTML

UNIT_006 : JQUERY — 06 HRS

JQUERY, HOW TO USE JQUERY?, Merits,methods,Introduction,History,What Is Typescript,Why We Use,Features,Ts & Ecmasccript,Components Of Ts,Declaration Files,Environment Setup,Basic Syntax,Compile & Exe ,Example,Classes,Modules

LAB SET FOR HTML ,CSS,BOOTSTRAP,AND JAVA SCRIPT

LAB 1 : HTML BASICS

- Create an HTML page with a header, footer, and navigation menu.
- Add different types of HTML elements such as headings, paragraphs, lists, and images.
- Create a simple form with input fields and a submit button.

LAB 2 : CSS STYLING

- Style the HTML elements using inline CSS, internal CSS, and external CSS.
- Apply different font styles, colors, and backgrounds to elements.
- Create a CSS-based layout for a web page using divs and CSS properties like float and margin.

LAB 3 : RESPONSIVE DESIGN WITH BOOTSTRAP

- Integrate Bootstrap into an HTML page using the Bootstrap CDN.
- Use Bootstrap classes to create responsive layouts and grids.
- Apply Bootstrap components such as navigation bars, buttons, and forms.

LAB 4 : CSS FLEXBOX AND GRID

- Use CSS Flexbox to create a flexible layout for a web page.
- Create a grid layout using CSS Grid to arrange elements in a structured manner.
- Combine Flexbox and Grid to create complex layouts.

LAB 5 : JAVASCRIPT BASICS

- Write JavaScript code to display a popup message or alert on a web page.
- Create a JavaScript function to validate a form before submitting it.
- Manipulate HTML elements dynamically using JavaScript.

LAB 6 : DOM MANIPULATION

- Use JavaScript to access and manipulate the Document Object Model (DOM) of a web page.
- Create interactive features like event listeners and handlers using JavaScript.
- Build a simple slideshow or image carousel using JavaScript.

LAB 7 : FORM VALIDATION WITH JAVASCRIPT

- Write JavaScript code to validate form input fields, such as checking for empty fields or valid email addresses.
- Display error messages dynamically based on user input using JavaScript.
- Implement client-side form validation before submitting data to the server.

LAB 8 : HANDLING EVENTS WITH JAVASCRIPT

- Write JavaScript code to handle different types of events, such as button clicks or mouse movements.
- Implement functionality to show and hide elements based on user actions using JavaScript event handlers.
- Use JavaScript to create interactive elements like dropdown menus or to

LAB 9 : AJAX AND API INTEGRATION

- Use JavaScript and AJAX to make asynchronous requests to a server and update web page content dynamically.
- Integrate an external API into a web page to fetch and display data using JavaScript.
- Implement functionality to search and filter data on a web page using AJAX and JavaScript.

LAB 10 : MINI PROJECT

Design and implement a simple web application that combines HTML, CSS, Bootstrap, and JavaScript.

ANGULAR

20 HRS

UNIT_007 : INTRODUCTION TO ANGULAR:

Understanding the fundamentals of Angular.
Overview of Angular architecture and its key features.
Setting up the development environment for Angular.

UNIT_008 : TYPESCRIPT:

Introduction to TypeScript, the language used in Angular development.TypeScript syntax, data types, variables, and functions.
Classes, interfaces, and modules in TypeScript.
TypeScript decorators and their usage in Angular.

UNIT_009 : ANGULAR COMPONENTS:

Understanding Angular components and their role in building applications.
Creating components using the Angular CLI (Command Line Interface).
Data binding and event handling in components.
Component lifecycle hooks and their usage.

UNIT_010 : ANGULAR TEMPLATES AND DIRECTIVES:

Angular template syntax and interpolation.
Working with built-in directives (e.g., ngIf, ngFor, ngSwitch).
Creating custom directives in Angular.

UNIT_011 : ANGULAR SERVICES AND DEPENDENCY INJECTION:

Creating and using services in Angular.
Understanding dependency injection and its importance.
Injecting services into components and other Angular constructs.
Sharing data and functionality across components using services.

UNIT_012 : ROUTING AND NAVIGATION:

Implementing client-side routing in Angular.
Configuring routes and route parameters.
Implementing nested routes and child components.
Using router guards for authentication and authorization.

UNIT_013 : ANGULAR FORMS:

Building forms in Angular using template-driven and reactive approaches.
Form validation and error handling.
Working with form controls and form groups.
Implementing custom form validators.

UNIT_014 : ANGULAR MODULES AND LAZY LOADING:

Understanding Angular modules and their role in organizing the application.
Creating and configuring modules in Angular.
Implementing lazy loading for optimizing application performance.

UNIT_015 : ANGULAR HTTP AND API INTEGRATION:

Making HTTP requests using the Angular HttpClient module.
Consuming RESTful APIs and handling responses.
Error handling and interceptors in HTTP requests.
Mocking HTTP requests for testing purposes.

UNIT_016 : ANGULAR TESTING AND DEBUGGING:

Writing unit tests for Angular components, services, and directives.
Using testing utilities and frameworks (e.g., Jasmine, Karma).
Debugging Angular applications using browser developer tools.
Performance optimization and debugging techniques.

UNIT_017 : ANGULAR DEPLOYMENT AND BUILD OPTIMIZATION:

Building and bundling Angular applications for production.
Optimizing application size and performance.
Deploying Angular applications to various hosting platforms.
Continuous integration and deployment (CI/CD) for Angular projects.

LAB SET ANGULAR

LAB 1 : SETTING UP ANGULAR DEVELOPMENT ENVIRONMENT

- Install Node.js and Angular CLI.
- Create a new Angular project using Angular CLI.
- Run and test the default Angular application in the browser.

LAB 2 : COMPONENTS AND TEMPLATES

- Create a new component and template.
- Use data binding to display dynamic content in the template.
- Implement event binding to handle user interactions.

LAB 3 : SERVICES AND DEPENDENCY INJECTION

- Create a service to handle data retrieval and manipulation.
- Inject the service into a component and use it to fetch and display data.
- Implement dependency injection to provide the service to the component.

LAB 4 : ROUTING AND NAVIGATION

- Configure routing in the Angular application.
- Create multiple components and associate them with different routes.
- Implement navigation between components using router links and programmatically.

LAB 5 : FORMS AND VALIDATION

- Create a form with input fields and submit functionality.
- Implement form validation using Angular's built-in validators.
- Display validation errors and provide user feedback.

LAB 6 : HTTP REQUESTS AND APIS

- Use Angular's HttpClient module to make HTTP requests to an API.
- Fetch data from an external API and display it in the application.
- Implement CRUD operations (create, read, update, delete) using HTTP requests.

LAB 7 : ANGULAR MATERIAL

- Install and configure Angular Material in the project.
- Use Angular Material components like buttons, cards, and forms.
- Apply styles and themes provided by Angular Material.

LAB 8 : STATE MANAGEMENT WITH NGRX

- Set up NgRx for state management in the Angular application.
- Create actions, reducers, and effects to manage application state.
- Use selectors to retrieve and display data from the store.

LAB 9 : AUTHENTICATION AND AUTHORIZATION

- Implement user authentication using a mock or real authentication service.
- Protect routes and components based on user authentication status.
- Restrict access to certain features based on user roles or permissions.

LAB 10 : DEPLOYMENT AND PRODUCTION BUILD

- Build the Angular application for production.
- Deploy the application to a hosting platform or server.
- Configure routing and server-side rendering for SEO optimization.

REACT

20 HRS

UNIT_018 : INTRODUCTION TO REACT:

Overview of React and its benefits.
Setting up the development environment for React.
Understanding the virtual DOM and React's rendering process.
Introduction to JSX (JavaScript XML) syntax.

UNIT_019 : COMPONENTS AND PROPS:

Understanding React components and their lifecycle.
Creating functional and class components.
Passing and accessing props in components.
Using state and managing component state.

UNIT_020 : JSX AND RENDERING ELEMENTS:

JSX syntax and its usage in React.
Rendering React elements to the DOM.
Conditional rendering and handling dynamic content.
Fragment and its usage for grouping multiple elements.

UNIT_022 : HANDLING EVENTS:

Adding event handlers to React components.
Binding event handlers and passing arguments.
Event delegation and event pooling in React.
Understanding synthetic events in React.

UNIT_023 : STATE MANAGEMENT WITH REACT:

Working with component state in React.
Updating state using setState and immutability concept.
Lifting state up and managing shared state.
Using controlled and uncontrolled components.

UNIT_024 : REACT ROUTER:

Implementing client-side routing using React Router.
Configuring routes and route parameters.
Navigating between different pages and components.
Implementing nested routes and route guards.

UNIT_025 : REACT FORMS:

Building forms in React.
Handling form inputs and managing form state.
Form validation and error handling.
Using form libraries and third-party form components.

UNIT_026 : MANAGING DATA WITH REACT:

Fetching and handling data from APIs.
Making HTTP requests using fetch or Axios.
Managing asynchronous operations with promises and async/await.
Using React hooks (e.g., useEffect, useState) for data management.

UNIT_027 : REACT CONTEXT API:

Understanding React's Context API for global state management.
Creating context providers and consumers.
Sharing data and state across components using context.
Implementing context with multiple providers and consumers.

UNIT_028 : REACT HOOKS:

Introduction to React hooks and their benefits.
Using useState for managing state in functional components.
useEffect hook for handling side effects and lifecycle events.
Custom hooks for reusable logic and code abstraction.

UNIT_029 : REACT TESTING AND DEBUGGING:

Writing unit tests for React components using testing libraries (e.g., Jest, React Testing Library).
Debugging React applications using browser developer tools and React DevTools.
Performance optimization and debugging techniques for React applications..

UNIT_030 : REACT DEPLOYMENT AND BUILD OPTIMIZATION:

Building and bundling React applications for production.
Optimizing application size and performance.
Deploying React applications to various hosting platforms.
Continuous integration and deployment (CI/CD) for React projects.

LAB SET REACT

LAB 1 :ETTING UP REACT DEVELOPMENT ENVIRONMENT

- Set up the development environment by installing Node.js and create-react-app.
- Create a new React project using create-react-app.
- Run and test the default React application in the browser.

LAB 2 : COMPONENTS AND JSX

- Create functional and class components in React.
- Use JSX syntax to render components and HTML elements.
- Pass props to components and display dynamic content.

LAB 3 : STATE AND LIFECYCLE METHODS

- Create stateful components and manage component state.
- Implement lifecycle methods such as componentDidMount and componentDidUpdate.
- Update component state and re-render components accordingly.

LAB 4 : HANDLING EVENTS

- Implement event handlers for user interactions such as button clicks or form submissions.
- Update component state based on user input.
- Use event objects and prevent default behavior when necessary.

LAB 5 : FORMS AND CONTROLLED COMPONENTS

- Create a form with input fields and handle form submissions.
- Implement controlled components by binding form input values to component state.
- Perform form validation and display error messages.

LAB 6 :LISTS AND KEYS

- Render lists of items using the map function.
- Assign unique keys to list items for efficient updates.
- Implement dynamic rendering of lists based on component state.

LAB 7 : CONDITIONAL RENDERING

- Conditionally render components or elements based on component state or props.
- Implement conditional rendering using if statements, ternary operators, or logical operators.

LAB 8 : REACT ROUTER

- Install and configure React Router in the project.
- Create routes and associate them with different components.
- Implement navigation between routes using links or programmatic navigation.

LAB 9 : CONTEXT API

- Create a context using React's Context API.
- Provide and consume context values in different components.
- Update context values and trigger re-renders in consuming components.

LAB 10 : HOOKS AND CUSTOM HOOKS

- Use built-in React hooks such as useState and useEffect in functional components.
- Create custom hooks to encapsulate reusable logic.
- Implement functionality like fetching data, handling side effects, or managing component state using hook

NODEJS BACKEND

UNIT_031 : 05 HRS

Introduction, Setup, Advantages And Disadvantages, Why Nodejs?, Nodejs , Architecture

UNIT_032 : 05 HRS

NodeJs Modules - Functions,Buffer,Module,Module Types, Core Modules,Local , Modules,Module.Exports

UNIT_033 : 07 HRS

(What Is NPM?Installing Packages Locally,Adding Dependency In Package.Json, Installing Packages Globally,Updating Packages, Debugging

UNIT_034 : 05 HRS

Creating Web Sever, Handling Http Methods And File System Reading, Writing, Opening And Deleting Files

UNIT_035 : 05 HRS

Event Driven Framework, Eventemitter Class, Inheriting Events,

UNIT_036 : 06 HRS

Express Framework , Installation, Middleware, How Does Express Work ? ,

UNIT_037 : 05 HRS

Error Handling, Routing, Handling Static Files,CORS,

UNIT_038 : 07 HRS

Database Integration with MySQL/MongoDV, Logging,

UNIT_039 :

CAPSTONE PROJECT

LAB SET NODEJS BACKEND

LAB 1 : SETTING UP NODE.JS ENVIRONMENT

- Install Node.js and set up the development environment.
- Create a basic Node.js project structure.
- Initialize a package.json file to manage project dependencies.

LAB 2 : BUILDING RESTFUL APIS WITH EXPRESS.JS

- Set up an Express.js application to handle HTTP requests.
- Define routes and endpoints for various CRUD operations.
- Implement middleware for request validation, authentication, and error handling.

LAB 3 : DATABASE INTEGRATION WITH MONGODB

- Install MongoDB and set up a local or remote database.
- Connect Node.js application to the MongoDB database.
- Perform CRUD operations on MongoDB collections using Mongoose.

LAB 4 : USER AUTHENTICATION AND AUTHORIZATION

- Implement user registration and login functionality.
- Use libraries like Passport.js for authentication strategies.
- Implement role-based access control for restricting access to certain routes.

LAB 5 : HANDLING FILE UPLOADS AND DOWNLOADS

- Handle file uploads from clients using libraries like Multer.
- Store uploaded files on the server or in cloud storage (e.g., AWS S3).
- Enable file downloads or streaming for clients.

LAB 6 : ERROR HANDLING AND LOGGING

- Implement error handling middleware to handle runtime errors.
- Set up logging using libraries like Winston or Morgan.
- Capture and log application errors and exceptions.

LAB 7 : TESTING AND TEST-DRIVEN DEVELOPMENT (TDD)

- Write unit tests using frameworks like Mocha or Jest.
- Implement test suites to cover different parts of the application.
- Practice test-driven development by writing tests before writing application code.

LAB 8 : AUTHENTICATION WITH JWT (JSON WEB TOKENS)

- Implement authentication using JWT for stateless authentication.
- Generate and verify JWT tokens for securing API routes.
- Use middleware to validate and decode JWT tokens.

LAB 9 : CACHING AND PERFORMANCE OPTIMIZATION

- Integrate caching mechanisms using libraries like Redis.
- Implement caching strategies to improve application performance.
- Optimize API response times and reduce database queries.

LAB 10 : DEPLOYMENT AND CONTINUOUS INTEGRATION

- Deploy Node.js application to cloud platforms like AWS or Heroku.
- Set up continuous integration and deployment pipelines using tools like Jenkins or Travis CI.
- Monitor and manage the deployed application using monitoring and logging tools.

THANK YOU

VISIT US @

+91 63635 07858

www.uplsnipe.com

D0WLOARD THE APP

GET IT ON Google Play

/uplsnipe

uplsnipe

#123, 70Th Cross, 2Nd Floor, SVA Arcade, 5Th Block Rajajinagara, Bengaluru - 560010