Front end Development Lectureflow

Module 1: Fundamentals of IT & Web Design	4

- Session 1: Understanding the IT and Web Ecosystem Trainer Coverage: What is IT? Overview of software and web careers. Distinction between Front-End, Back-End, Full Stack, UI/UX Design. How a website request travels from browser ? server ? database ? browser. Introduction to static vs dynamic websites. Depth: Students should be able to explain how clicking a link loads a page and identify at least three major web career roles. Demo: Open a website (e.g., Netflix) ? use Chrome DevTools ? sh
- Session 2: Internet Fundamentals & Web Hosting Trainer Coverage: How the Internet works (IP, DNS, HTTP/HTTPS, Domain Name System). Client-Server Model: Browser ? Web Server ? Database. Difference between local server (XAMPP, WAMP) and hosting server. Depth: Students must conceptually understand that a domain name points to an IP and a web server serves files. Demo: Use nslookup google.com (Windows command prompt) or DNS lookup tool. Show live domain-hosting purchase simulation (GoDaddy)
- Session 3: Role of a Web Designer Trainer Coverage: Job responsibilities: Layout design, UI/UX basics, responsive development, testing, and optimization. Introduction to design thinking (Empathize ? Define ? Ideate ? Prototype ? Test). Tools overview: VS Code, Figma, Canva, Uizard, GitHub. Depth: Students should grasp that web design isn't just aesthetics — it involves usability, accessibility, and performance. Demo: Show a Figma design ? explain color hierarchy, typography, layout consisten
- Session 4: Web Standards, Workflow & Version Control Trainer Coverage: W3C compliance, cross-browser standards, semantic structure. What is version control and why it's essential (intro to Git & GitHub). Overview of workflow: Design ? Code ? Test ? Deploy ? Maintain. Depth: Students should understand why HTML validation, CSS standards, and Git tracking matter in professional teams. Demo: Create a GitHub repo, push an "index.html" file, show commit history.

Module 2) WD - HTML	10

- Session 1: Introduction to HTML Trainer Coverage: What is HTML, syntax, and document structure (<!DOCTYPE>, <html>, <head>, <body>). Basic tags: headings, paragraphs, comments, horizontal lines. Depth: Students must write valid HTML from scratch. Demo: Create a sample "My First Webpage" using VS Code.
- Session 2: Working with Links and Images Trainer Coverage: Anchor tags <a href>, image tags <img src>. Relative vs Absolute paths. Alt attributes and SEO relevance. Demo: Insert images and link them to external pages. Depth: Students must understand accessibility importance of alt text.
- Session 3: Lists and Tables Trainer Coverage: Ordered, Unordered, and Definition lists. Table structure: <table>, <tr>, <th>, <td>, <caption>. Styling table borders with inline CSS. Demo: Create a "Course Schedule Table" for a web design bootcamp.
- Session 4: Forms & Inputs Trainer Coverage: <form> structure, method=get/post, input types (text, password, email, radio, checkbox, select). Form validation attributes (required, pattern).

Demo: Create a "Contact Us" form. Depth: Students must understand difference between form front-end validation and server-side handling

- Session 5: Semantic HTML5 Tags Trainer Coverage: Semantic structure: <header>, <nav>, <article>, <aside>, <section>, <footer>. Importance of SEO and screen readers. Demo: Convert a <div>-based layout into semantic tags. Depth: Explain difference between <div> layout and semantic sectioning.

- Session 6: Multimedia Integration Trainer Coverage: Embedding video (<video>), audio (<audio>), and YouTube iframes. Attributes: autoplay, controls, loop. Demo: Add a company intro video to homepage.

- Session 7: Details & summary, image mapping, create progressbar of skill (create Html page that show your personal detail, skill using progress bar ,project details using details & summary)

- Session 9: Structuring a Complete Page Trainer Coverage: Building multi-section pages (header, nav, content, footer). Internal linking between pages (about.html, contact.html). Demo: Develop a simple "Personal Portfolio" layout.

- Session 8: Accessibility & SEO-Friendly HTML Trainer Coverage: Role of alt, title, aria-label. Meta tags: description, keywords, viewport. Proper heading structure for search engines. Demo: Use W3 Validator to check for missing meta tags.

- Session 10: Integration and Validation Trainer Coverage: Testing responsiveness using Chrome DevTools. HTML Validation using W3C Validator. Practical: Each student validates their HTML and corrects errors. Depth: Emphasize professionalism — "No validation errors = publish-ready code."

Module 3) WD - HTML5 - Project	1

- Build a 2–page HTML5 website including navigation, profile details, and contact form. Instructors should review for semantic structure and accessibility compliance. AI Add-on: Use ChatGPT to write the bio and generate meta tags automatically.

Module 4) WD - CSS and CSS 3	8

- Session 1 – Introduction to CSS and Selectors Coverage: What is CSS and how it separates structure from style. Types: Inline, Internal, External. Syntax: selector { property: value }. Basic selectors (id, class, element), grouping selectors. Demo: Apply color, font size to paragraphs and headings. Depth: Students must link external CSS properly and understand cascading priority.

- Session 2 – Colors, Backgrounds & Typography Coverage: RGB, HEX colors, Gradients. Font-family, weight, spacing, line-height. Google Fonts integration. Demo: Design a "Hero Section" banner with gradient background and custom fonts.

- Session 3 – Box Model & Positioning Coverage: Margin, Padding, Border, Box-Sizing. Static, Relative, Absolute, Fixed, Sticky positions. Demo: Create a "Sticky Navigation Bar." Depth: Explain browser rendering flow and layout impact.

- Session 4 – Display Properties & Layout Coverage: Block, Inline, Inline-Block, None. Float & Clearfix. Demo: Two-column layout using float. Depth: Show why floats were replaced by Flexbox.

- Session 5 – Borders, Shadows, and Backgrounds Trainer Coverage: border-radius, box-shadow, text-shadow, background-image. Demo: Design a modern "card" element with shadow and background.
- Session 6 – Lists, Tables, and Forms Styling Trainer Coverage: Styling lists and tables with CSS. Customizing form inputs and buttons. Demo: Style a contact form with hover effects.
- Session 7 – CSS Specificity & Inheritance Trainer Coverage: CSS cascade rules (inline > internal > external). Inheritance hierarchy and specificity calculation. Demo: Conflict resolution exercise — "Why didn't my color apply?"
- Session 8 – Mini Project: Styled Portfolio Page Deliverable: Recreate HTML portfolio page from Module 2 with full CSS styling (text, colors, sections). AI Add-on: Use ChatGPT to generate gradient color schemes for different sections.

Module 5) WD – Advance CSS/CSS Preprocessors	10

- Session 1: CSS3 Enhancements Overview Topics to Cover: CSS3 new features vs legacy CSS2. Rounded corners, shadows, gradients. Attribute selectors ([type="text"]). Depth: Students must identify visual effects only possible with CSS3. Practical Demo: Create cards with rounded corners and drop shadows.
- Session 2: Transitions Topics to Cover: Transition syntax: property, duration, timing-function, delay. Hover transitions on buttons and images. Depth: Explain user experience benefits of smooth transitions. Demo: Fade color change on hover with .transition class.
- Session 3: Transformations Topics to Cover: transform functions: rotate, scale, skew, translate. Combining multiple transforms. Demo: Create an image zoom effect on hover.
- Session 4: CSS Animations Topics to Cover: @keyframes declaration. Animation properties: name, duration, iteration-count, direction. Demo: Create a "bouncing ball" animation. Depth: Explain how browser repaints frames over time.
- Session 5: Flexbox Deep Dive Topics to Cover: display:flex, justify-content, align-items, flex-wrap. Aligning elements horizontally/vertically. Demo: Responsive navbar using Flexbox. AI Integration: Use ChatGPT to draw Flexbox layout diagram.
- Session 6: CSS Grid System Topics to Cover: Defining rows & columns with grid-template. grid-area naming, gap property. Demo: Design a responsive image gallery. Depth: Explain difference between Flexbox (1D) vs Grid (2D).
- Session 7: Pseudo-Classes and Elements Topics to Cover: :hover, :focus, :nth-child(), ::before, ::after. Styling link states. Demo: Create a custom list with icons via pseudo-elements.
- Session 8: Mini Project – Animated Homepage Section Goal: Design a hero banner with animation and Flexbox layout. AI Add-on: Use Copilot to auto-suggest animation keyframes.

Module 6) WD - Framework	11

- Session 1: Introduction to Bootstrap Framework Topics to Cover: What is Bootstrap, why it's used. Linking via CDN. Container, row, and column basics. Demo: Create a two-column layout using Bootstrap.
- Session 2: Bootstrap Typography & Components Topics to Cover: Headings, tables, buttons, lists. Cards and utility classes. Demo: Build a "Services" section using Bootstrap cards.

- Session 3: Navbar & Carousel Topics to Cover: Navbar structure and customization. Carousel component setup and controls. Demo: Design a homepage header with navbar + image slider.
- Session 4: Forms & Validation Topics to Cover: Bootstrap form elements, floating labels. Validation states and messages. Demo: Create registration form with validation feedback.
- Session 5: Utility Classes Topics to Cover: Margin/padding (m-3, p-2), display, colors. Visibility and alignment utilities. Demo: Responsive spacing adjustments using utility classes.
- Session 6: Tailwind CSS Introduction Topics to Cover: Linking Tailwind via CDN. Utility-first concept. Using Tailwind for layout and color utilities. Demo: Rebuild a Bootstrap hero section with Tailwind.
- Session 7: Responsive Design Concepts Topics to Cover: Mobile-first design philosophy. Breakpoints and media queries. Accessibility (viewport meta tag, ARIA labels). Demo: Adjust site layout for 3 screen sizes (mobile/tablet/desktop).
- Session 8: Mini Project – Responsive Landing Page Goal: Build a responsive business homepage with navbar, services, and contact form. AI Add-on: Use Figma AI to generate layout ? translate into Bootstrap HTML.

Module 8) WD - JQuery Basic, Effects & Advanced	5

- Session 1: Introduction to jQuery Topics to Cover: Why use jQuery? Linking CDN. Selectors: $(selector).action(). Demo: Hide and show elements with buttons.
- Session 2: Effects & Animations Topics to Cover: .hide(), .show(), .fadeIn(), .slideToggle(). Animation chaining. Demo: Create expanding "Read More" section.
- Session 3: DOM Manipulation & Events Topics to Cover: Changing text, CSS, and HTML dynamically. Handling click, keyup, change events. Demo: Create live word counter in textarea.
- Session 4: AJAX Introduction Topics to Cover: What is AJAX, its purpose. Fetching JSON from a mock API. Demo: Load user data dynamically without reloading page. AI Integration: Use ChatGPT to show how same AJAX logic converts to modern fetch() syntax.
- Session 5: Mini Project – Live Search Bar Goal: Implement search suggestion feature using jQuery + JSON. Trainer Tip: Relate this to real-world search boxes like Amazon.

Module 9) JavaScript Essentials And Advanced	20

- Session 1: Introduction to JavaScript & Environment Setup Coverage Depth: What JavaScript is (client-side vs server-side). How browsers interpret JS and connect it with HTML via <script> tag. Running JS in console (browser + VS Code Live Server). Demo: Add alert("Welcome to your Bio-Link App!") inside an HTML file. Learning Goal: Students understand how JS executes and connects to DOM.
- Session 2: Variables & Data Types Coverage Depth: var, let, const—scope & reassignment. Primitive vs Reference types. Dynamic typing in JS. Demo: Create variables: let userName="Aarya", let followers=1500. Display in console. Link to App: Represent profile details in JS variables.
- Session 3: Operators & Conditions Coverage Depth: Arithmetic, Logical, Comparison, and Ternary operators. Truthy/Falsy values concept. Demo: If followers > 1000 ? show "Verified Creator ??". Trainer Tip: Relate conditional checks to real user-verification logic (like Instagram).

- Session 4: Loops & Iteration Coverage Depth: For, While, Do-While, For-of, ForEach. Difference between indexed and iterable loops. Demo: Loop through an array of social links and print each in console. Learning Goal: Understand repetition and dynamic rendering concept.
- Session 5: Functions & Reusability Coverage Depth: Function declaration vs expression vs arrow function. Parameters and return values. Demo: Function createButton(label,color) that returns a styled HTML button.
- Session 6: DOM Manipulation – Selecting Elements Coverage Depth: getElementById, querySelector, innerHTML, style. Reading & updating content dynamically. Demo: Add new link element dynamically from input form. Trainer Note: Show real-time DOM change in browser.
- Session 7: DOM Events & Interactivity Coverage Depth: Event types: onclick, onchange, addEventListener. Event object basics. Demo: "Add Link" button creates new card instantly. Depth: Explain event flow concept (intro only).
- Session 8: Arrays & Objects – Data Structures Coverage Depth: Array creation, methods (push, pop, splice). Object literals and nested objects. Demo: let links=[{title:"Instagram",url:"..."},...] Loop to display all cards. Learning Goal: Represent app data as structured collections.
- Session 9: JSON & LocalStorage (Persistence Basics) Coverage Depth: Converting data: JSON.stringify() ? JSON.parse(). Using localStorage.setItem/getItem/removeItem. Demo: Save links to localStorage; reload page = data persists. Trainer Note: Relate this to database persistence later (JDBC concept preview).
- Session 10: ES6 Syntax & Modern Features Coverage Depth: Template literals, default params, destructuring, spread/rest. Arrow functions recap. Demo: console.log(\Hello ${userName}, you have ${followers} followers!`)`. Depth: Use modern syntax in all subsequent examples.
- Session 11: Asynchronous JS – Callbacks, Promises, Async/Await (Part 1) Coverage Depth: Blocking vs non-blocking code. What are callbacks? Creating and resolving Promises. Demo: Simulate loading message using setTimeout.
- Session 14: Modular JavaScript (Imports & Exports) Coverage Depth: File modularization with export & import. Default vs named exports. Demo: Move link-creation functions to utils.js and import in main file. Depth: Explain benefit for scalability (pre-React architecture).
- Session 12: Asynchronous JS (Part 2) – Fetch API Coverage Depth: fetch() GET & POST. Chaining Promises & handling responses. Parsing JSON responses. Demo: Fetch mock user data from https://jsonplaceholder.typicode.com/users. Depth: Show network tab request ? response cycle.
- Session 15: Classes & OOP in JS Coverage Depth: Declaring classes, constructors, methods. Creating instances. Demo: class UserProfile { constructor(name, followers){...} } Add method displayProfile(). Trainer Note: Show parallel with Java classes ? OOP bridge for Phase 4.
- Session 13: Error Handling in JS Coverage Depth: try…catch…finally, custom errors, throw. Importance of defensive coding. Demo: If URL input invalid ? throw custom Error ("Invalid URL"). AI Integration: ChatGPT explains error stack trace for student practice.
- Session 16: Advanced Array Methods Coverage Depth: map(), filter(), reduce(), find(), some(), every(). Real use cases for transformation. Demo: Filter active links (isActive:true), calculate total clicks. AI Tool: Ask ChatGPT to suggest one-liner map-reduce alternatives.
- Session 17: Event Delegation & Dynamic UI Updates Coverage Depth: Event bubbling vs capturing. Delegating events for dynamic elements. Demo: Click "X" icon ? remove card without re-

binding listeners. Depth: Students must grasp efficiency benefit.

- Session 18: Debugging & Developer Tools Coverage Depth: Console methods (log, warn, error, table). Setting breakpoints in Sources tab. Understanding stack traces. Demo: Debug addLink() and inspect variables. AI Tip: Ask ChatGPT to explain error messages from console.
- Session 19: Mini-Project Preparation & Integration Coverage Depth: Combine HTML, CSS, and JS into one interactive application. Plan modules: Form ? Data Storage ? Display ? Edit/Delete. Activity: Sketch architecture on whiteboard (show how data flows between functions). Trainer Goal: Help students map concepts to app features clearly.
- Session 20: Integration Testing & Polish Coverage Depth: Manual QA: test each button and form field. Common bugs & fixes (null values, double adds). Basic UI validation patterns. Demo: Final run-through of Social Bio Link App.

Module-10) React - Fundamentals	14

- Session 1 – Introduction to React Topics to Cover: What is React? Why React (Virtual DOM, Components). create-react-app setup. Demo: Run first React app and modify default component.
- Session 2 – JSX & Components Topics to Cover: JSX syntax and rules. Function vs Class components. Demo: Render "Hello {username}" component.
- Session 3 – Props Topics to Cover: Passing data between components. Default props and prop types. Demo: Reusable "Card" component with props.
- Session 4 – State & useState Hook Topics to Cover: State management basics. Updating state. Demo: Counter increment/decrement app.
- Session 5 – Event Handling Topics to Cover: Handling click, input, form events. Demo: Interactive form capturing user data.
- Session 6 – Lists & Conditional Rendering Topics to Cover: Rendering arrays with map(). Conditional rendering (if, ternary). Demo: Render dynamic product list.
- Session 7 – useEffect Hook Topics to Cover: Component lifecycle and side effects. Demo: Simulate API fetch on mount.
- Session 8 – useRef & Controlled Forms Topics to Cover: Managing DOM elements using refs. Demo: Focus input field after submit.
- Session 9 – Routing Topics to Cover: React Router setup. Routes, NavLink, BrowserRouter. Demo: Multi-page navigation (Home, About, Contact).
- Session 10 – Context API Topics to Cover: Global state management. useContext hook. Demo: Theme toggle (dark/light mode).
- Session 11 – Axios & API Integration Topics to Cover: GET and POST using Axios. Demo: Fetch product list from mock API.
- Session 12 – Error Handling Topics to Cover: Try/catch in API calls. Demo: Display "Error loading data" message.
- Session 13 – Deployment Topics to Cover: npm run build, Netlify deployment. Demo: Publish React SPA online.
- Session 14 – Mini Project: React Portfolio Builder Goal: Build portfolio website with reusable components. AI Add-on: Use Copilot to suggest component structure.

Module 11) React Advanced (Hooks, Firebase, Redux)	20

- Session 1 – Introduction to Advanced React Concepts Topics to Cover: Review: props, state, useEffect, component hierarchy. Why scalability matters (component reusability, maintainability). Introduction to advanced hooks (custom hooks, reducer hooks). Depth: Explain problems with prop drilling and redundant code in large apps. Practical Demo: Show an example of repetitive API call logic to motivate custom hooks.
- Session 2 – Custom Hooks (Part 1) Topics to Cover: What is a custom hook and naming conventions (useSomething). Converting logic from components into hooks. Rules of hooks (only call at top level). Demo: Build useFetchData() hook for fetching API data. Depth: Explain how custom hooks promote reusability across components.
- Session 3 – Custom Hooks (Part 2) Topics to Cover: Passing parameters to custom hooks. Managing multiple useEffects inside a custom hook. Returning data and loading/error states. Demo: Enhance useFetchData() to include loading spinner and error handling.
- Session 4 – useReducer Hook (Part 1) Topics to Cover: Concept of reducers (state + action ? new state). useReducer() syntax and initialization. Demo: Build counter example with increment/decrement/reset using useReducer. Depth: Explain the difference between useState vs useReducer for complex state logic.
- Session 5 – useReducer Hook (Part 2) Topics to Cover: Multiple action handling (switch case). Dispatching from child components. Demo: To-Do app using reducer for add, edit, delete actions.
- Session 6 – useMemo & useCallback Hooks Topics to Cover: React re-rendering and performance bottlenecks. Memoization with useMemo(). Function memoization with useCallback(). Demo: Create large list rendering with and without memoization; compare performance.
- Session 7 – Context API Deep Dive (Part 1) Topics to Cover: What is global state and why needed. Creating Context and Provider. Passing state to deeply nested components. Demo: Create ThemeContext for global dark/light mode. Depth: Explain prop drilling problem and how Context solves it.
- Session 8 – Context API Deep Dive (Part 2) Topics to Cover: Combining Context with Reducer. Nested context providers. Avoiding unnecessary re-renders using memoization. Demo: User authentication context using reducer + context.
- Session 9 – Firebase Introduction Topics to Cover: What is Firebase and its role (backend as a service). Firebase setup: project creation, SDK integration. Environment variable management for Firebase keys. Demo: Initialize Firebase project and connect to React app. Depth: Explain benefits of serverless integration for front-end developers.
- Session 10 – Firebase Authentication (Part 1) Topics to Cover: Sign-up, login with email/password. Firebase Auth API overview. Demo: Build SignUp and Login components using Firebase methods.
- Session 11 – Firebase Authentication (Part 2) Topics to Cover: Persistent authentication with onAuthStateChanged(). Logout and user session handling. Demo: Implement logout functionality with user greeting in navbar. Depth: Emphasize secure handling of user session states.
- Session 12 – Firestore CRUD Operations (Part 1) Topics to Cover: Firestore introduction (collections, documents). Add and read data. Demo: Add and fetch tasks in a "Task Manager" app.
- Session 13 – Firestore CRUD Operations (Part 2) Topics to Cover: Update and delete documents. Real-time updates with onSnapshot(). Demo: Real-time task updates without refreshing. Depth: Explain how Firebase syncs UI state automatically.

- Session 14 – Redux Fundamentals Topics to Cover: Redux architecture: store, action, reducer, dispatch. Setting up Redux in a React project. Demo: Create simple counter using Redux store.
- Session 15 – Redux State Flow Topics to Cover: Connecting components with useSelector and useDispatch. Redux DevTools overview. Demo: Display and update tasks using Redux actions. Depth: Show difference between Context and Redux for large-scale apps.
- Session 16 – Async Operations in Redux (Thunk) Topics to Cover: Middleware concept. Async API calls using Redux Thunk. Demo: Fetch task list from Firebase using thunk action.
- Session 17 – Form Handling with Formik Topics to Cover: Controlled vs uncontrolled components. Installing and using Formik. Validation with Yup schema. Demo: Task creation form with validation. Depth: Highlight cleaner syntax and validation vs manual handling.
- Session 18 – Environment Variables & Configuration Topics to Cover: Environment management (.env.local). API key security. Conditional environments (dev vs prod). Demo: Secure Firebase credentials in environment files.
- Session 19 – Mini Project Build: "Task Manager App" (Part 1) Topics to Cover: Create base structure: authentication + CRUD logic. Redux integration for task management. Demo: Build and test authentication + task list UI. AI Tip: Use Copilot to assist in creating CRUD reducers and components.
- Session 20 – Mini Project Deployment & Review Topics to Cover: Firebase Hosting or Netlify deployment. Testing, debugging, performance optimization. Final code walkthrough. Demo: Deploy the Task Manager app live and share link. AI Integration: Ask ChatGPT: "Generate README.md for this React + Firebase project."

Module-12) GraphQL & Next.js	**10**

- Session 1 – Introduction to GraphQL Topics to Cover: What is GraphQL and how it differs from REST APIs. Key concepts: Schema, Query, Mutation, Resolver. Benefits: Fewer API calls, strongly typed queries. Depth: Students must understand how GraphQL allows frontends to request only the data they need. Practical Demo: Use the public GraphQL Pokémon API in GraphiQL Playground — run a simple query to fetch names and images.
- Session 2 – GraphQL Setup in React Topics to Cover: Setting up Apollo Client. Installing dependencies: @apollo/client and graphql. Connecting to GraphQL endpoint. Demo: Configure ApolloProvider and test a simple query to display user names. Depth: Explain the Apollo Provider's role as React Context for GraphQL data.
- Session 3 – Writing Queries and Displaying Data Topics to Cover: Basic useQuery hook usage. Fetching and rendering data dynamically. Handling loading and error states. Demo: Build "User Directory" fetching list of users with names and emails. Depth: Explain destructuring of { loading, error, data } and conditional rendering.
- Session 4 – Mutations (Add, Update, Delete) Topics to Cover: Writing GraphQL mutations. Using the useMutation hook in Apollo. Optimistic UI updates. Demo: Add a new user entry using mutation and update UI instantly. Depth: Show how mutation modifies backend and automatically re-renders the component.
- Session 5 – GraphQL Schema & Playground Practice Topics to Cover: Understanding schema structure and types. Custom queries and relationships. Using Apollo Sandbox/GraphiQL for testing. Demo: Test nested queries and learn introspection. Depth: Students must understand what type

definitions mean (type User { id: ID! name: String! }).

- Session 6 – Introduction to Next.js Topics to Cover: What is Next.js, and why use it over Create React App. SSR (Server-Side Rendering) vs CSR (Client-Side Rendering). Next.js folder structure overview. Demo: Create first Next.js app using npx create-next-app. Depth: Explain how SSR improves SEO and load performance.
- Session 7 – Next.js Pages and Routing Topics to Cover: File-based routing: /pages directory. Dynamic routes [id].js. Navigation with Link component. Demo: Build 3-page blog app (Home, About, Post Detail). Depth: Clarify automatic routing and static path generation. AI Add-on:
- Session 9 – Authentication and Environment Variables Topics to Cover: Adding Firebase or NextAuth.js for authentication. Environment files in Next.js (.env.local). Protecting API routes. Demo: Add Google login to Next.js app. Depth: Teach handling protected routes with session validation.
- Session 8 – Data Fetching in Next.js Topics to Cover: getStaticProps, getServerSideProps, getStaticPaths. When to use each method. Combining with GraphQL queries. Demo: Fetch and render blog posts at build time using getStaticProps. Depth: Explain difference between SSR and SSG with diagrams.
- Session 10 – Mini Project: "AI-Powered Blog Platform" Goal: Build a fully functional blog platform integrating: Next.js for SSR + Routing GraphQL API for content Authentication (Firebase/NextAuth) OpenAI API for AI-generated article summaries

Module 13) AI Tools for Front-End Developers	6

- Session 1 – The Role of AI in Modern Web Design Topics to Cover: What is generative AI and its place in web development. Overview of AI tools: ChatGPT, Copilot, Firefly, Figma AI, Uizard. Ethical AI usage and prompt engineering basics. Depth: Students should understand how AI assists developers (not replaces them). Demo: Use ChatGPT to generate a sample landing page structure (HTML outline).
- Session 2 – Using ChatGPT & GitHub Copilot for Coding Topics to Cover: How to use ChatGPT for debugging, explaining errors, and refactoring. Installing and using Copilot inside VS Code. Writing effective prompts for code generation. Depth: Emphasize that AI helps accelerate, not automate without review. Demo: Refactor a React component using Copilot auto-suggestions. AI Exercise: Ask ChatGPT: "Optimize this React component for cleaner code and reusable hooks."
- Session 3 – Figma AI and Uizard for Design Automation Topics to Cover: Using Figma AI to generate wireframes. Uizard overview – text-to-design prototypes. Converting Figma designs to HTML/CSS. Demo: Type a text prompt in Uizard ("Create a fitness app homepage") ? export design preview. Depth: Discuss design-to-code handoff workflows. AI Tip: Show Figma plugin "Figma to Code" in action for live export.
- Session 4 – AI Image & Asset Generation Topics to Cover: Using Adobe Firefly / Leonardo.ai for image creation. Generating icons, backgrounds, and hero images using prompts. Image optimization (size and format). Demo: Generate a "modern tech startup hero image" using Firefly ? import to project. Depth: Discuss prompt tone (lighting, color, realism). AI Integration: Show differences between Midjourney and Firefly results.
- Session 5 – SEO & Content with AI Topics to Cover: Creating meta titles, keywords, and alt text using ChatGPT. Generating blog posts and product descriptions. Avoiding plagiarism and bias.

Demo: Generate SEO tags for a blog post in ChatGPT ? integrate into HTML <head>. Depth: Discuss importance of keyword clustering for organic reach. AI Tip: Ask: "Generate SEO title, description, and alt text for an e-commerce landing page."

- Session 6 – Mini Project: "AI-Generated Landing Page" Goal: Build a fully AI-assisted website prototype. Steps: Use Uizard or Figma AI for layout. Generate hero images via Firefly. Write content (headlines, text) using ChatGPT. Code the layout in HTML/CSS/React. AI Integration: Copilot for code generation, ChatGPT for copywriting and bug explanation.

Module-14) Capstone Project	10

- Session 1 – Project Orientation & Idea Finalization Topics to Cover: Explain project scope and goals. Brainstorm ideas and finalize concept. Example Project Themes: AI Resume Builder (React + Firebase + OpenAI API) Smart E-Commerce Dashboard (Next.js + GraphQL) AI Blog Creator (React + Firebase + ChatGPT content) AI Integration: Use ChatGPT to brainstorm project features and flow diagrams.
- Session 2 – Wireframing & UI Design Topics to Cover: Low-fidelity vs high-fidelity wireframes. Figma AI / Uizard prototyping. Demo: Create a wireframe for homepage and dashboard UI. Depth: Discuss user journey mapping and layout hierarchy. AI Tip: Ask Figma AI: "Generate layout for job profile dashboard."
- Session 3 – Environment Setup & Boilerplate Creation Topics to Cover: Next.js / React app setup. Folder structure for scalability. Environment variables for APIs. Demo: Create .env.local for OpenAI or Firebase keys. AI Integration: Ask ChatGPT: "Generate secure .env configuration for my project."
- Session 4 – Component Development (Frontend) Topics to Cover: Navbar, Footer, Cards, Forms, Modals. Props and State flow. Demo: Build dynamic navigation + form component. Depth: Focus on reusable, modular component design. AI Tip: Use Copilot for scaffolding repetitive UI components.
- Session 5 – Backend Integration (Firebase or GraphQL) Topics to Cover: Setting up Firestore or GraphQL API. CRUD operations integration. Demo: Connect form to Firestore to store and fetch data. AI Integration: Ask ChatGPT: "Refactor this API call with async/await and better error handling."
- •Redux Core concepts – Actions , Reducers , Store React-Redux
- Session 6 – Authentication & State Management Topics to Cover: Firebase Auth / NextAuth.js. Redux or Context API for session management. Demo: Implement login + user dashboard with persisted session. Depth: Emphasize UI reactivity to login state.
- Session 7 – AI API Integration Topics to Cover: Using OpenAI API or Hugging Face API. Fetching AI-generated content dynamically. Demo: Generate blog summary or resume content from user input using OpenAI API. AI Add-on: Show how prompt tuning changes output quality.
- Session 8 – Responsive Design & Testing Topics to Cover: Testing on mobile, tablet, desktop using DevTools. Browser compatibility checks. Lighthouse performance optimization. Demo: Run Google Lighthouse and fix issues. AI Integration: Ask ChatGPT: "Suggest improvements for Lighthouse performance report."
- Session 9 – Final Deployment Topics to Cover: Hosting options: Firebase Hosting, Netlify, or Vercel. Connecting custom domain. Demo: Deploy project and verify live link. Depth: Discuss CI/CD basics briefly for future learning.

- Redux - Complexity of Managing state - Understand the Redux Flow - Setting up Reducer and store - Dispatching Actions - Passing and Retrieving Data with Action - Combining Multiple Reducers - Adding Middleware - Redux Dev tools