

KAUR'S CODING ACADEMY

Teen Web and App Developer - Grades 9 to 12

Module-wise Teacher Syllabus

A structured teen developer roadmap covering modern frontend, JavaScript, React, APIs, backend basics, GitHub and deployment.

Course	Audience	Suggested Duration	Final Output
Teen Web and App Developer - Grades 9 to 12	Grades 9 to 12	8 to 12 months / career starter pathway	Portfolio website, API app, React project, backend API and full-stack capstone

Teacher Note

Introduce professional habits early: clean folders, reusable code, debugging, GitHub commits, deployment links and project explanation.

Learning Outcomes

- Create responsive websites with strong HTML, CSS and JavaScript fundamentals.
- Build React applications using components, props, state, hooks and routing.
- Work with APIs, JSON, async/await and error handling.
- Understand backend APIs, databases, authentication and deployment basics.
- Build and present a full-stack or advanced frontend capstone project.

Document Use

This PDF is made for teachers to plan classes module by module. Each module includes topic names, teaching points, class practice and revision work.

Complete Module Index

The table below gives a quick view of what will be taught in this course.

Module	Topic Name	Main Things Teacher Will Cover
1	Web Architecture and Developer Tools	How browser, server and database work together, VS Code setup, Browser developer tools, Project folder structure, and more
2	HTML5 Professional Structure	Semantic tags, Forms and validation attributes, Media elements, Tables, and more
3	CSS3 Responsive UI Design	Selectors and specificity, Box model, Flexbox, Grid, and more
4	JavaScript Core Programming	Variables, data types and operators, Strings and numbers, Arrays and objects, Functions and scope, and more
5	Conditions, Loops and Problem Solving	If/else and switch, Logical operators, For, while and for-of loops, Break and continue, and more
6	DOM, Events and Browser Apps	DOM selection, Changing content and styles, Events and event listeners, Form handling, and more
7	APIs, JSON and Async JavaScript	JSON data format, Fetch and Axios basics, Promises, Async/await, and more
8	Git, GitHub and Documentation	Git init/add/commit/status, GitHub repository, README file, Commit messages, and more
9	React Setup and First Application	React purpose, Single page applications, Vite/React app setup, Project structure, and more
10	React Components, Props and State	Functional components, Reusable components, Props and children, useState, and more
11	React Rendering, Lists and Forms	Conditional rendering, List rendering and keys, Controlled inputs, Form validation, and more
12	React Hooks and Effects	useState recap, useEffect, useRef, useMemo and useCallback basics, and more
13	React Router and Multi-page Apps	React Router setup, Routes and links, Nested routes, Route params, and more
14	Backend and REST API Basics	Client-server model, HTTP methods, REST endpoints, Request and response, and more
15	Database and CRUD Concepts	Database purpose, SQL vs NoSQL overview, Collections/tables, CRUD operations, and more
16	Authentication and Security Basics	Signup and login flow, Password hashing idea, JWT concept, Protected routes, and more
17	Deployment and Project Hosting	GitHub Pages basics, Netlify/Vercel frontend deployment, Backend hosting overview, Environment variables, and more
18	Capstone Project and Demo Preparation	Project planning, Feature list, Wireframes, Frontend-backend integration, and more

Detailed Module-wise Teaching Plan

Teachers should first explain the concept in simple language, then demonstrate code, then allow students to practice and customize the output.

Module 1: Web Architecture and Developer Tools

Topics to be Covered	What Teacher Will Explain / Demonstrate
- How browser, server and database work together - VS Code setup - Browser developer tools - Project folder structure - Basic terminal commands	- Explain client-server flow with diagrams - Demonstrate editor extensions and browser inspection - Show how to create clean project folders
Class Practice / Student Output	Homework / Revision
- Set up development environment - Create first project folder - Inspect a website using browser tools	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 2: HTML5 Professional Structure

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Semantic tags - Forms and validation attributes - Media elements - Tables - Accessibility basics - SEO-friendly meta tags	- Show why semantic HTML matters - Demonstrate accessible labels and alt text - Explain page structure used in real websites
Class Practice / Student Output	Homework / Revision
- Build a multi-section portfolio page - Add contact form and media - Check semantic structure	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 3: CSS3 Responsive UI Design

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Selectors and specificity - Box model - Flexbox - Grid - Transitions - Animations - Media queries	- Explain layout systems with real UI sections - Demonstrate responsive navigation and cards - Show debugging layout issues
Class Practice / Student Output	Homework / Revision
- Create a responsive landing page - Add animated services section - Test on phone and desktop widths	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 4: JavaScript Core Programming

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Variables, data types and operators - Strings and numbers - Arrays and objects - Functions and scope - ES6 syntax	- Explain JS as problem-solving language - Demonstrate data manipulation in console - Show reusable functions for repeated logic
Class Practice / Student Output	Homework / Revision
- Complete logic challenge set - Build calculator functions - Create object-based student data	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 5: Conditions, Loops and Problem Solving

Topics to be Covered	What Teacher Will Explain / Demonstrate
- If/else and switch - Logical operators - For, while and for-of loops - Break and continue - Algorithm practice	- Teach dry run method - Demonstrate flowcharts to code - Show debugging with console logs
Class Practice / Student Output	Homework / Revision
- Solve grade checker, fizz buzz and pattern challenges - Explain one solution in class	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 6: DOM, Events and Browser Apps

Topics to be Covered	What Teacher Will Explain / Demonstrate
- DOM selection - Changing content and styles - Events and event listeners - Form handling - Timers and date/time - localStorage	- Explain DOM as page control through JS - Demonstrate interactive UI changes - Show how localStorage saves data
Class Practice / Student Output	Homework / Revision
- Build quiz app or expense tracker - Add validation and saved data - Improve user messages	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 7: APIs, JSON and Async JavaScript

Topics to be Covered	What Teacher Will Explain / Demonstrate
- JSON data format - Fetch and Axios basics - Promises - Async/await - Loading states - Error handling	- Explain API request-response flow - Demonstrate calling a public API - Show handling success, loading and error states
Class Practice / Student Output	Homework / Revision
- Build weather app or movie search app - Display API data in cards - Handle failed searches	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 8: Git, GitHub and Documentation

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Git init/add/commit/status - GitHub repository - README file - Commit messages - Project screenshots and links	- Explain why version control matters - Demonstrate pushing a project to GitHub - Show a good README structure
Class Practice / Student Output	Homework / Revision
- Create GitHub repository - Push a web project - Write a simple README	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 9: React Setup and First Application

Topics to be Covered	What Teacher Will Explain / Demonstrate
- React purpose - Single page applications - Vite/React app setup - Project structure - JSX rules	- Explain component-based development - Demonstrate project creation and running server - Show difference between HTML and JSX
Class Practice / Student Output	Homework / Revision
- Create first React app - Edit App component - Run and test locally	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 10: React Components, Props and State

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Functional components - Reusable components - Props and children - useState - State update patterns	- Explain reusable UI building blocks - Demonstrate parent-to-child data passing - Show how state changes UI
Class Practice / Student Output	Homework / Revision
- Build profile cards - Create counter and product card components - Pass data using props	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 11: React Rendering, Lists and Forms

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Conditional rendering - List rendering and keys - Controlled inputs - Form validation - Reusable form components	- Explain rendering decisions - Demonstrate mapping arrays to UI - Show form state handling
Class Practice / Student Output	Homework / Revision
- Build registration form - Create dynamic product list - Show validation messages	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 12: React Hooks and Effects

Topics to be Covered	What Teacher Will Explain / Demonstrate
- useState recap - useEffect - useRef - useMemo and useCallback basics - Custom hooks introduction	- Explain when effects run - Demonstrate API call in useEffect - Show ref for input focus
Class Practice / Student Output	Homework / Revision
- Build API dashboard - Add loading and error state - Create one small custom hook	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 13: React Router and Multi-page Apps

Topics to be Covered	What Teacher Will Explain / Demonstrate
- React Router setup - Routes and links - Nested routes - Route params - Default and error pages	- Explain routing in single page apps - Demonstrate navigation bar - Show dynamic detail pages with params
Class Practice / Student Output	Homework / Revision
- Build multi-page React website - Add product detail route - Add 404 page	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 14: Backend and REST API Basics

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Client-server model - HTTP methods - REST endpoints - Request and response - Postman testing	- Explain how frontend talks to backend - Demonstrate GET and POST API examples - Show testing with Postman
Class Practice / Student Output	Homework / Revision
- Create simple API endpoint - Test with Postman - Connect one frontend request	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 15: Database and CRUD Concepts

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Database purpose - SQL vs NoSQL overview - Collections/tables - CRUD operations - Basic schema design	- Explain how apps store data - Show task/user/product data model - Demonstrate create/read/update/delete flow
Class Practice / Student Output	Homework / Revision
- Design database fields for project - Create sample CRUD plan - Discuss validation rules	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 16: Authentication and Security Basics

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Signup and login flow - Password hashing idea - JWT concept - Protected routes - Role-based access overview	- Explain authentication vs authorization - Demonstrate token-based login flow conceptually - Show why passwords should not be stored directly
Class Practice / Student Output	Homework / Revision
- Plan login feature for capstone - Create protected route checklist - Write user roles and access rules	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 17: Deployment and Project Hosting

Topics to be Covered	What Teacher Will Explain / Demonstrate
- GitHub Pages basics - Netlify/Vercel frontend deployment - Backend hosting overview - Environment variables - Common deployment errors	- Show deployment steps for frontend - Explain build command and output folder - Demonstrate checking live link
Class Practice / Student Output	Homework / Revision
- Deploy one frontend project - Share live URL - Fix common broken path issue	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 18: Capstone Project and Demo Preparation

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Project planning - Feature list - Wireframes - Frontend-backend integration - Testing - Presentation and viva questions	- Guide students through milestone planning - Review code and UI quality - Train students to explain architecture and features
Class Practice / Student Output	Homework / Revision
- Complete capstone project - Prepare GitHub and live demo link - Present final project	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Final Review and Certification Checklist

- Student can explain the main concepts in simple words.
- Student has completed the required practice tasks and mini projects.
- Student can run the project, fix common errors and present the work confidently.
- For advanced courses, student should submit GitHub repository, screenshots and deployment link where possible.