

KAUR'S CODING ACADEMY

Real-Time Websites, Apps and Games - Project Guide

Module-wise Teacher Syllabus

A project guide for teachers to help students plan, build, test, improve and present real-world websites, apps and games.

Course	Audience	Suggested Duration	Final Output
Real-Time Websites, Apps and Games - Project Guide	All project batches: school, college and professional learners	Use with any course / 4 to 8 week project sprint	Portfolio-ready website, app or game with documentation and presentation

Teacher Note

Use this guide after students learn fundamentals. Do not start with a big project before students can build small features independently.

Learning Outcomes

- Plan real-world project ideas with problem statement and feature list.
- Build websites, apps and games using the right technology for the learner level.
- Apply UI design, logic, API, data storage, testing and deployment practices.
- Prepare project documentation, screenshots and final presentation.
- Create a project portfolio suitable for parents, schools, colleges, clients or employers.

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	Project Planning and Problem Statement	Choosing website/app/game project idea, Problem statement, Target users, Feature list, and more
2	Wireframe and UI Planning	Rough sketch, Page/screen list, Navigation flow, Color and typography planning, and more
3	Real-Time Website Projects	Portfolio website, Restaurant/cafe website, School/event website, Business landing page, and more
4	Real-Time Web App Projects	Calculator app, To-do app, Expense tracker, Quiz app, and more
5	Real-Time Mobile App Projects	News reader app, Fitness tracker, Task management app, Food order app, and more
6	Backend and API Project Features	User signup/login, CRUD operations, Database schema, REST API endpoints, and more
7	Game Project Concepts	Game goal, Player controls, Enemies/obstacles, Score and timer, and more
8	Data Storage and User Progress	localStorage for browser projects, AsyncStorage for mobile projects, Database storage for full-stack projects, Saving scores/profiles/tasks, and more
9	Testing, Debugging and Quality Review	Testing checklist, Common UI bugs, Common logic bugs, Console and error messages, and more
10	Deployment and Sharing	GitHub repository, README file, Screenshots, Frontend deployment, and more
11	Project Presentation	Demo flow, Explaining problem and solution, Showing features, Explaining technologies used, and more
12	Assessment Rubric and Portfolio Readiness	UI design quality, Code structure, Feature completion, Bug-free behavior, 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: Project Planning and Problem Statement

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Choosing website/app/game project idea - Problem statement - Target users - Feature list - Project scope - Success criteria	- Explain why projects need planning - Demonstrate problem-to-feature mapping - Help students avoid too many features at once
Class Practice / Student Output	Homework / Revision
- Write project title and problem statement - List 5 to 8 features - Define final demo goal	- 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: Wireframe and UI Planning

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Rough sketch - Page/screen list - Navigation flow - Color and typography planning - User-friendly layout - Mobile responsiveness	- Explain wireframe before coding - Show simple homepage/app screen sketch - Discuss spacing and readability
Class Practice / Student Output	Homework / Revision
- Draw wireframe - Create screen/page list - Approve final layout plan	- 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: Real-Time Website Projects

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Portfolio website - Restaurant/cafe website - School/event website - Business landing page - E-commerce frontend - LMS/course website - Blog/news website	- Explain which features make a website real-world - Demonstrate common sections: hero, services, gallery, contact - Show responsive and SEO-friendly structure
Class Practice / Student Output	Homework / Revision
- Select one website project - Build homepage sections - Add contact form and responsive layout	- 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: Real-Time Web App Projects

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Calculator app - To-do app - Expense tracker - Quiz app - Weather app - Movie search app - Notes app - Dashboard app	- Explain difference between website and web app - Demonstrate input-processing-output flow - Show storage or API integration
Class Practice / Student Output	Homework / Revision
- Build one app with interactive logic - Add localStorage or API - Improve error messages and UI	- 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: Real-Time Mobile App Projects

Topics to be Covered	What Teacher Will Explain / Demonstrate
- News reader app - Fitness tracker - Task management app - Food order app - Study planner app - Habit tracker app - Chat UI prototype	- Explain mobile screen flow - Show navigation and reusable components - Discuss offline data and backend needs
Class Practice / Student Output	Homework / Revision
- Create screen flow - Build core screens - Connect sample API/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 6: Backend and API Project Features

Topics to be Covered	What Teacher Will Explain / Demonstrate
- User signup/login - CRUD operations - Database schema - REST API endpoints - Validation - Authentication and authorization - Postman testing	- Explain backend features used in real apps - Demonstrate API endpoint planning - Show how frontend and backend connect
Class Practice / Student Output	Homework / Revision
- Design API routes - Create database model - Test APIs in Postman	- 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: Game Project Concepts

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Game goal - Player controls - Enemies/obstacles - Score and timer - Collision detection - Levels and difficulty - Restart and game over	- Explain games as rules and state changes - Demonstrate movement and score logic - Help students balance fun and difficulty
Class Practice / Student Output	Homework / Revision
- Build memory, snake, maze, quiz battle or typing game - Add score and restart - Test game 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 8: Data Storage and User Progress

Topics to be Covered	What Teacher Will Explain / Demonstrate
- localStorage for browser projects - AsyncStorage for mobile projects - Database storage for full-stack projects - Saving scores/profiles/tasks - Data privacy basics	- Explain where project data should be stored - Compare local storage and database - Demonstrate saving and loading user progress
Class Practice / Student Output	Homework / Revision
- Add saved data feature - Test data after refresh/restart - Document stored fields	- 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: Testing, Debugging and Quality Review

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Testing checklist - Common UI bugs - Common logic bugs - Console and error messages - Form validation testing - Responsive testing	- Teach students to test like users - Demonstrate debugging workflow - Review UI, functionality and code cleanliness
Class Practice / Student Output	Homework / Revision
- Complete testing checklist - Fix at least 3 bugs - Ask peer/parent to test 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.

Module 10: Deployment and Sharing

Topics to be Covered	What Teacher Will Explain / Demonstrate
- GitHub repository - README file - Screenshots - Frontend deployment - Backend deployment overview - Live demo link - Common deployment errors	- Explain why deployed links matter - Demonstrate GitHub and deployment workflow - Show how to write project description
Class Practice / Student Output	Homework / Revision
- Push project to GitHub - Deploy where possible - Update README with screenshots and features	- 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: Project Presentation

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Demo flow - Explaining problem and solution - Showing features - Explaining technologies used - Challenges faced - Future improvements	- Train students to speak clearly and confidently - Ask project-based questions - Help students prepare final demo script
Class Practice / Student Output	Homework / Revision
- Prepare 2 to 3 minute demo - Present project - Answer teacher questions	- 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: Assessment Rubric and Portfolio Readiness

Topics to be Covered	What Teacher Will Explain / Demonstrate
- UI design quality - Code structure - Feature completion - Bug-free behavior - Responsiveness - Documentation - Presentation confidence	- Explain grading criteria before assessment - Review project against rubric - Give final improvement plan
Class Practice / Student Output	Homework / Revision
- Submit final project folder/repo - Complete rubric checklist - Add project to portfolio	- 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.