Web Development Mini Project Ideas PDF

A web development mini project should take one to four weeks and teach one layer well: layout with HTML and CSS, interactivity with JavaScript, an API-driven React app, or a small Node.js and database back end. A portfolio site, weather app, expense tracker and URL shortener are good choices.

Key takeaways

- Start with HTML, CSS and JavaScript before frameworks.
- Use free public APIs to make apps feel real.
- Make every project mobile friendly.
- Host it free on GitHub Pages, Netlify or Vercel.
- Each idea fits in one to four weeks.

Who this PDF is for

This PDF is for 2nd and 3rd year students who want web projects for labs, semester submissions or a first portfolio. The ideas start with plain HTML, CSS and JavaScript, then add React, APIs and a small back end. Each plan lists features, steps and what to explain in the viva.

How to choose your project

- **Match your syllabus**: pure front end if you have not studied back end yet.
- **Write a feature list** before you code, and stop when it is done.
- **Test on a phone** as well as a laptop.
- **Put it on GitHub** with a short README.

Beginner mini projects (1 to 2 weeks)

1. Personal Portfolio Website

- **Problem:** Show your profile, skills and projects.
- **Data and services:** Your own content.
- **Tech stack:** HTML, CSS (Flexbox and Grid), a little JavaScript.
- **Main modules:** Header, about, projects grid, contact form, dark mode.
- **Steps:** Sketch the layout, build mobile first, add a contact form with a form service, deploy.
- **Show in the viva:** How media queries make it responsive.

- **Extensions:** Add a blog section.

2. Weather App

- **Problem:** Show current weather and forecast for a city.
- **Data and services:** A free weather API such as OpenWeatherMap (free key).
- **Tech stack:** HTML, CSS, JavaScript (fetch).
- **Main modules:** Search box, API call, result card, error handling, unit toggle.
- **Steps:** Get an API key, fetch by city name, show results, handle invalid cities.
- **Show in the viva:** Why the API key should not be exposed in a public repo.
- **Extensions:** Add a 5-day chart.

3. Quiz App

- **Problem:** Multiple-choice quiz with score and timer.
- **Data and services:** Questions you write in a JSON file.
- **Tech stack:** HTML, CSS, JavaScript.
- **Main modules:** Question loader, timer, scoring, results page.
- **Steps:** Load questions, show one at a time, track score and time, show a summary.
- **Show in the viva:** How the app keeps state.
- **Extensions:** Store high scores in localStorage.

4. To-do List with Local Storage

- **Problem:** Add, complete and delete tasks that persist.
- **Data and services:** No external data.
- **Tech stack:** HTML, CSS, JavaScript.
- **Main modules:** Task form, list rendering, filters, storage.
- **Steps:** Build add, edit, delete and filter, save to localStorage on each change.
- **Show in the viva:** How localStorage works and its limits.
- **Extensions:** Add drag-and-drop sorting.

Intermediate mini projects (2 to 3 weeks)

5. Expense Tracker with Charts

- **Problem:** Track income and expenses with category charts.
- **Data and services:** No external data.
- **Tech stack:** React, Chart.js.
- **Main modules:** Transaction form, list, filters, monthly chart, CSV export.
- **Steps:** Build components, manage state, draw charts by category, export CSV.
- **Show in the viva:** Props and state in React.
- **Extensions:** Add a Node.js back end.

6. Movie Search App

- **Problem:** Search movies and save favourites.
- **Data and services:** A free movie API such as OMDb or TMDB (free key).

- **Tech stack:** React, React Router.
- **Main modules:** Search, results grid, details page, favourites.
- **Steps:** Fetch on search, debounce input, route to a details page, save favourites.
- **Show in the viva:** What debouncing does.
- **Extensions:** Add infinite scroll.

7. URL Shortener

- **Problem:** Turn long links into short codes with click counts.
- **Data and services:** No external data.
- **Tech stack:** Node.js, Express, MongoDB or SQLite.
- **Main modules:** Short code generator, redirect route, click counter, simple UI.
- **Steps:** Generate unique codes, store mappings, redirect and count clicks.
- **Show in the viva:** How you avoid duplicate codes.
- **Extensions:** Add QR codes and expiry dates.

8. Blog with Markdown Editor

- **Problem:** Write and publish posts in Markdown.
- **Data and services:** No external data.
- **Tech stack:** Node.js, Express, EJS or React, a database.
- **Main modules:** Editor, preview, posts list, categories, comments.
- **Steps:** Build CRUD for posts, render Markdown safely, add comments.
- **Show in the viva:** How you prevent script injection in user content.
- **Extensions:** Add login for authors.

Stretch mini projects (3 to 4 weeks)

9. Real-time Chat App

- **Problem:** Chat rooms with live messages.
- **Data and services:** No external data.
- **Tech stack:** Node.js, Socket.IO, React or plain JavaScript.
- **Main modules:** Rooms, usernames, live messages, typing indicator.
- **Steps:** Set up Socket.IO, join rooms, broadcast messages, show who is typing.
- **Show in the viva:** WebSockets versus normal HTTP requests.
- **Extensions:** Store history in a database.

10. Student Result Portal

- **Problem:** Students view marks; admin uploads results.
- **Data and services:** Sample results you create.
- **Tech stack:** PHP and MySQL, or Node.js and MySQL.
- **Main modules:** Admin login, CSV upload, student lookup, mark sheet print view.
- **Steps:** Design tables, parse the CSV upload, show results by roll number.
- **Show in the viva:** How you prevent SQL injection.
- **Extensions:** Add a PDF mark sheet.

11. Recipe Finder

- **Problem:** Find recipes by ingredients you have.
- **Data and services:** A free recipe API such as TheMealDB.
- **Tech stack:** React or Vue.
- **Main modules:** Ingredient input, search, results, recipe page, shopping list.
- **Steps:** Call the API by ingredient, show matches, build a shopping list.
- **Show in the viva:** How you handle slow or failed API calls.
- **Extensions:** Add filters for vegetarian dishes.

12. Kanban Task Board

- **Problem:** Move tasks across To do, Doing and Done columns.
- **Data and services:** No external data.
- **Tech stack:** React, a drag-and-drop library, Firebase or a Node.js API.
- **Main modules:** Boards, columns, cards, drag and drop, sync.
- **Steps:** Build columns and cards, add drag and drop, save to the back end.
- **Show in the viva:** How state updates after a drag.
- **Extensions:** Add team sharing.

Mini project report structure

A mini project report is usually 10 to 20 pages. Keep it short and clear:

1. **Title page and certificate** in your department's format.
2. **Introduction:** the problem in one paragraph and your objectives as a list.
3. **Tools used:** languages, frameworks, editor, hosting service and any APIs.
4. **Design:** one block diagram or flowchart, plus page wireframes and, if you have a back end, the database tables.
5. **Implementation:** what each module does, with 3 to 6 screenshots.
6. **Results:** screenshots on desktop and mobile, and the live link.
7. **Conclusion and future scope,** then references.

Ask your guide for the exact format before you start writing.

Common mistakes to avoid

- Desktop-only layouts that break on phones.
- API keys committed to GitHub.
- Copying a tutorial without adding your own feature.
- No README or live link.

Frequently asked questions

What is a good web development mini project for beginners?

A portfolio website, weather app or quiz app. They teach layout, JavaScript and APIs without a back end.

Should I use React for a mini project?

Only after you are comfortable with plain JavaScript. React suits intermediate projects such as an expense tracker or movie search app.

Where can I host my mini project for free?

GitHub Pages, Netlify and Vercel host front-end projects free. Some platforms also offer free tiers for small back ends.

How do I make my mini project different?

Add one feature of your own, make it mobile friendly and deploy it with a clear README.

Does this PDF include full source code?

No. It gives the problem, data, stack, modules and steps for each project. You write the code yourself, which is what examiners test in the viva.

Is this PDF free?

Yes. Fill in the short form on this page and the PDF downloads right away.

Sources and further reading

- MDN Web Docs: Learn web development
- React documentation
- Node.js documentation
- Socket.IO documentation

About this PDF

Prepared by the Uncodemy training team as practice project ideas. Datasets and tools named are public; check each dataset's licence before use. We update it every January. Corrections: info@uncodemy.com.

More free resources and courses: uncodemy.com | info@uncodemy.com | +91 98183 66550