



Session 01 — Setting Up Your Trivia Game

Welcome to React Development! 🎉

You're about to launch your dev setup and start building like a pro. This guide breaks everything down into bite-sized tasks so you can move fast, learn by doing, and see results right away. Ready to dive in? Let's go!

Table of Contents

- [Launch Your Codespace](#)
- [Start the Development Server](#)
- [Replace the Placeholder Component](#)
- [Update the Page Title](#)
- [Ask the AI](#)



Launch Your Codespace

Let's get your cloud coding environment up and running so you can start building without messing with local installs. This is your dev playground in the cloud — no setup headaches, no installs, just code.



Steps

1. Head over to github.com and log in using the account you set up during pre-camp.
2. Pick one of these two ways to launch your Codespace:
 - **Quick link method:** [Just click this link](#)
 - **Manual method:**
 - Go to github.com/wizcamp/wizcamp-realms-demo
 - Click **Use this template** → **Open in codespace** in the top-right corner of the page.
3. Wait a bit while your Codespace builds. You'll see a **VS Code** editor pop up in your browser.
4. Once it loads, make sure you can see the project files in the file explorer on the left.

5. Dark mode not your thing? Click the gear icon in the bottom left, go to **Themes -> Color Theme**, and pick your favorite.

Why This Matters

Codespaces give everyone the same setup — no more “it works on my machine” drama. When you launch a Codespace, it spins up a fresh cloud environment with all the tools and dependencies you need pre-installed, so you can focus on building. Best part, if you mess something up, just delete and start fresh. You will use this Codespace for all sessions.

Bonus Challenge

Visit github.com/codespaces to explore more about managing your Codespaces.

Peek inside the `.devcontainer/devcontainer.json` file to see how your Codespace is configured — it’s like a blueprint for your dev environment.

Start the Development Server

Preview the app in your browser by running the dev server to confirm everything is wired up correctly.

Steps

1. Launch the dev server from the terminal: `npm run dev`.
2. Click “Open in Browser” or visit the provided localhost URL (e.g., `http://localhost:5173/`).
3. The starter app should load, showing a placeholder component.

Why This Matters

A running dev server lets you see and test your app as you build it. It gives you instant feedback on your code changes and allows you to interact with the app, speeding up your development workflow.

Bonus Challenge

Visit github.com/codespaces to explore more about managing your Codespaces.


Replace the Placeholder Component

With the development server still running, replace the placeholder component with the game’s splash screen component to experience live updates in action.

Steps

1. Open `src/App.jsx`
2. Add the import: `import SplashScreen from './components/SplashScreen';`
3. Replace `<StartHere />` with `<SplashScreen />`
4. Watch the screen update instantly — no save needed!

Why This Matters

Components are the building blocks  of **React** web apps — kind of like digital LEGO pieces. You build apps by snapping them together.

You probably noticed the `.jsx` file extension. That's because these components are written in a special syntax called **JSX**. It looks a lot like HTML, but is actually JavaScript under the hood. JSX lets you describe what the UI should look — using a syntax that's readable like HTML but powered by JavaScript.

The live update “magic” you experienced is actually powered by a build tool we are using called **Vite**, which uses a process known as **Hot Module Replacement (HMR)** to apply “smart updates” to your app instantly as you code.

Update the Page Title

Even though React apps are built with components, they still use a standard HTML file as the entry point. Let's update the page title to reflect our project name.

Steps

1. Open `index.html`
2. Identify where the HTML head/title is defined
3. Update the `title` tag to the name of our project:
`Wizcamp Realms - Legends of Trivia`
4. Confirm the browser tab displays the new title.

Why This Matters

A descriptive page title is important for usability, accessibility, and SEO. It helps users identify your app when they have multiple tabs open and improves discoverability in search engines.

Ask the AI — What Just Happened?

You just launched your Codespace, ran your dev server, swapped a component, and updated your page title — nice work! 🎯

Now let's make sure you understand what you did and why it matters. Use these AI prompts to dig deeper into the tools and concepts you just used. You can ask your AI assistant any of these questions or use them as a starting point to explore further.

🧠 AI Prompts That Reinforce Session 01 Tasks

🔧 Codespaces & Setup

- “What is a Codespace and why are we using it?”
- “How does a Codespace work behind the scenes?”
- “What does the `.devcontainer.json` file do?”
- “Why is cloud development better for beginners?”

💻 VS Code Basics

- “How do I open and edit files in VS Code?”
- “How do I change the theme in VS Code?”
- “What is the file explorer and how do I use it?”

🚀 Running the Dev Server

- “What does `npm run dev` do?”
- “What is a development server and why do we need it?”
- “What does the localhost URL mean?”
- “Why does the app reload when I change code?”

⚡ React + Vite Basics

- “What is Vite and why is it used in this project?”
- “How does Hot Module Replacement work?”
- “What is JSX and how is it different from HTML?”
- “Why do React components use `.jsx` files?”

🧩 Component Swapping

- “How do I import a component in React?”
- “What does `import SplashScreen from './components/SplashScreen'` mean?”
- “How do I replace one component with another?”
- “Why does the screen update instantly when I change components?”

HTML & Page Title

- “How do I change the page title in a React app?”
 - “Where is the `index.html` file and what does it do?”
 - “Why is the title tag important for websites?”
-

Pro Tip:

You don't have to memorize everything — just ask your AI assistant when you're curious or stuck. Think of it as your trusty AI sidekick, always ready to help.