

ReactJS Challenge - Build a Visual coding environment

Looking for expert ReactJS developers with a sense of aesthetics! About Juspay:

Our vision is to enable the billion people in India with frictionless payments experience & extend it with the right credit offerings. We power the payments experiences for 100M+ and have 200M+ SDK installs across top mobile apps in India.

In addition to our primary business, we contribute to the ecosystem by participating in early system design, protocol creation & reference implementations for country level initiatives. Examples are - building the core components of UPI, the BHIM app, protocol creation for OCEN (open credit enablement network) and Beckn (an open hyperlocal commerce platform).

To do this, our people need to be 10X effective. Such a culture is in the air. Freshers create frameworks to solve these big problems. Our framework Presto is a small size react native equivalent. We have fully embraced functional programming which is the new trend that is taking over the world.

Platform & People innovation to solve country level problems like never before.

Problems - BIG <ul style="list-style-type: none">- Payments<ul style="list-style-type: none">- Transport Rail/Bus, Rail, Metro,- Organized Retail, e-Comm- HyperLocal, Unorganized Retail- B2B small, B2B large businesses- GSTN- Hyperlocal<ul style="list-style-type: none">- Order, Delivery, Tracking platform- Smart Auto ordering - IOT- Smart Cities<ul style="list-style-type: none">- Data driven governance - IOT- Trust & Sharing, Credit platform	Platform - 10y ahead <ul style="list-style-type: none">- UI & Flow Component Framework/IDE<ul style="list-style-type: none">- Payments, Ordering, Tracking- Rapid Dev - 1 Week apps- Auto Quality check platform- Analytics Platform<ul style="list-style-type: none">- Conversions, A/B testing- Auto Anomaly detection- Trust/Credit Decisioning- Reverse engineering Tools- Infra<ul style="list-style-type: none">- Juspay Private cloud,- Bank / Govt - On Premise- Infra in a Box	People - Nurture for 10x <ul style="list-style-type: none">- Engineering Learning track<ul style="list-style-type: none">- Functional Programming- Systems Thinking- Data Science- IOT, 3D printing- Product learning<ul style="list-style-type: none">- Learn to code- Strategy, Storytelling, Growth hacking- Design Thinking, Prototyping- Design learning<ul style="list-style-type: none">- Deconstructing, Deep Arts learning - Sketching, Sculpture, Photo/Movie, Music
Space <p>Engineers, Designers, Data Analysts, Product Strategist, Product Makers, Evangelists/Storytellers Warriors, Explorers, Connectors/Enablers Deep Work spaces, Education Spaces, Design Thinking/Collaboration area, IOT/3D Printing Lab, Music/Video/Art Studio, Health Bar</p>		

Why React:

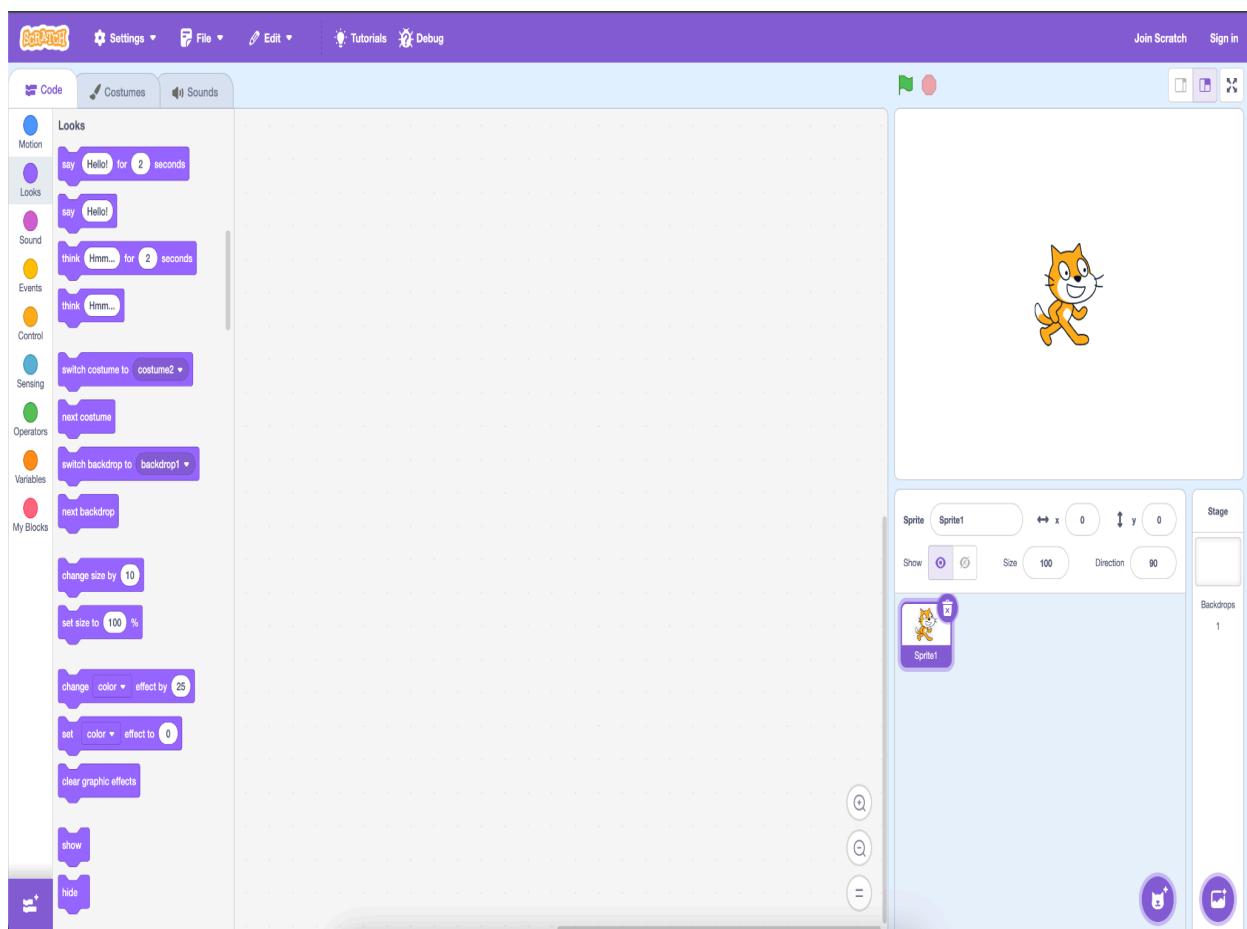
React has been inspired by functional programming aficionados, who brought those good practices to the Javascript ecosystem. In fact, the react's creator Jordan Walke

talks about how deeper FP technologies like ReScript is going to help react further
https://www.youtube.com/watch?v=5fG_lyNuEAw

We see React + FP as a way to build **complex operations dashboards, immersive data visualisation and creation/configuration tools** to be used by our customers, product managers, and business operations teams. These dashboards should be highly interactive, highly usable for repeat use cases & should facilitate frequent switching across desktop and mobile seamlessly.

About the challenge:

Build a visual code editor for Javascript that is similar to MIT Scratch <https://scratch.mit.edu/projects/editor/>



Challenge Instructions:

- Objective: Implement the following features within the provided project:

1. Motion Animations:

- a. Build the following animations under the ‘Motion’ category:
 - i. Move ____ steps
 - ii. turn ____ degrees
 - iii. Go to x: ____ y: ____
 - iv. Repeat animation (inside controls)

2. Looks Animations:

- a. Build the following animations under the “Looks” category:
 - i. Say ____ for ____ seconds
 - ii. think ____ for ____ seconds

- b. Ensure all corresponding drag-and-drop functionalities, as seen in the Scratch app, are covered.

3. Multiple Sprites Support:

- a. Add functionality to create multiple ‘Sprites.’
- b. Ensure the animations from Feature 1 are available for each Sprite.
- c. Implement a play button on click of which all Sprites starts animating

4. **Hero Feature** - Collision-Based Animation Swap:

- a. Introduce a "Hero Feature" that swaps animations when two characters collide.
- b. Example:
 - i. Character 1: [Move 10 steps, repeat animation]
 - ii. Character 2: [Move -10 steps, repeat animation]
 - iii. After collision, the animations swap:
 1. Character 1: Move -10 steps
 2. Character 2: Move 10 steps
- c. This feature adds dynamic interaction between characters in the Scratch playground.

- **Note:**

- It's mandatory to add the hero feature.
- Only build the animations mentioned above.
- Prefer using ReScript React; alternatively, JavaScript/TypeScript is acceptable.
- Use TailwindCSS if possible.
- Hero features are not included in the Scratch example or provided video. Contact us if you have any questions.

Project Setup

A sample base project has been created for you to build the features on top of that. Follow the below instructions to setup and get started

1. Go to the below link and clone the repo

<https://bitbucket.org/shiva-nandan/scratch-starter-project/src/master/>

(The above link is for demo visualization)

2. Run `npm i` to install dependencies

3. Run `npm start` and open `http://localhost:3000` to see the app