Frontend Development with React.js

Project Documentations

1.Introduction

Project Title: InsightStream: Navigate the News Landscape

Team members:

Register number	Name
C24UG104CAP028	R.Rayesha
C24UG104CAP019	M.Mahashree
C24UG104CAP021	M.Manjula
C24UG104CAP022	P.Megala
C24UG104CAP024	K.Pavatharani

2. Project Overview

Purpose: Briefly describe the purpose and goals of the project.

Features: Highlight the key features and functionalities of the frontend.

- The project is envisioned as more than a series of tasks; it is a journey of transformation that aligns innovation with meaningful impact. At its core, the initiative seeks to bridge gaps between current challenges and future opportunities, creating a stream of insights that guide decisions with clarity and precision.
- By integrating diverse perspectives, leveraging data, and fostering collaboration, the project aims to unlock value that extends beyond immediate deliverables. Every phase, from planning to execution, is seen not as an isolated step but as part of a
- continuous flow where lessons learned shape the next direction. This stream-like approach ensures adaptability, resilience, and sustained momentum, ultimately steering the project toward outcomes that are both practical and visionary.

```
npx create-react-app news-insight-stream
cd news-insight-stream
 npm install axios react-router-dom3. Architecture
```

3.Component Structure:

Outline the structure of major React components and how they interact.

- State Management: Describe the state management approach (e.g., Context API, Redux).

- Routing: Explain the routing setup (e.g., React Router).

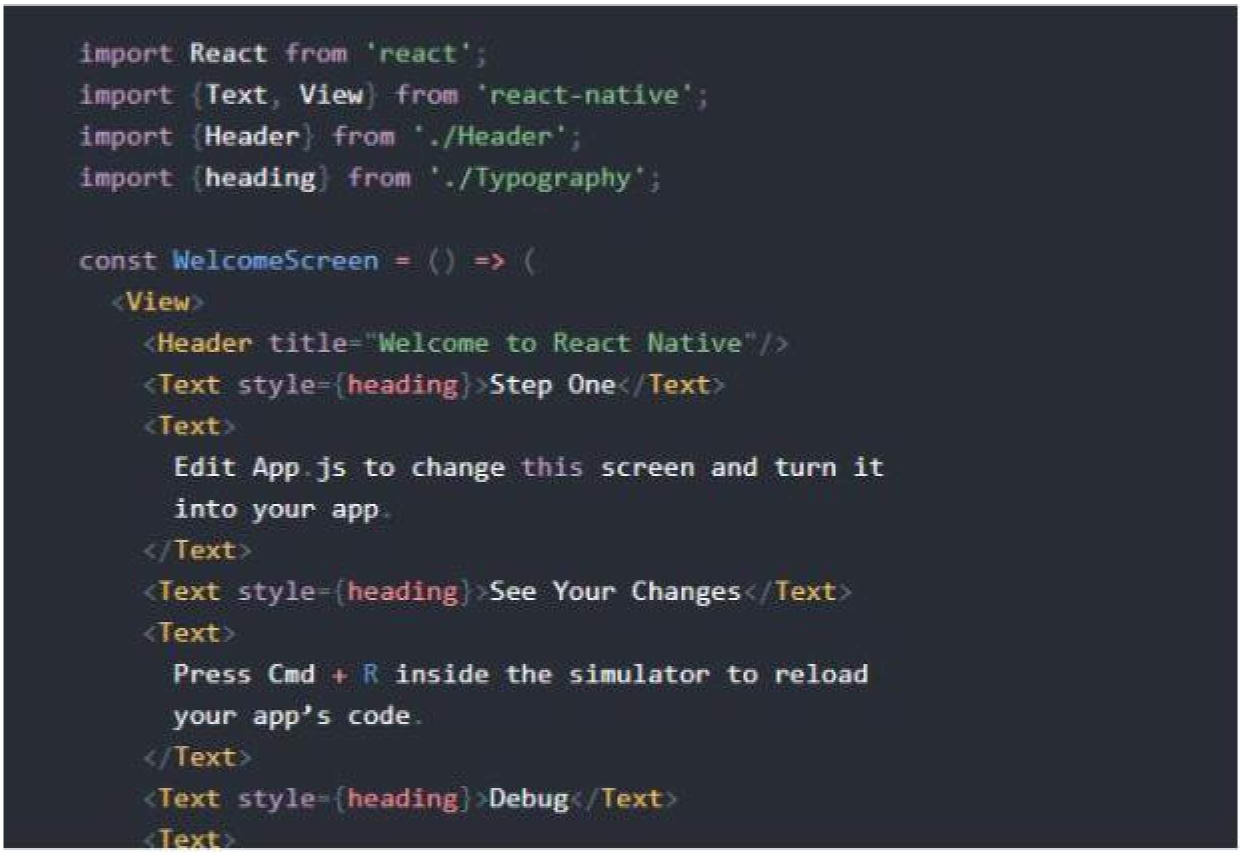
NewsList.jsx	Displays news items
NewsItem.jsx	Single news card
Filters.jsx	Category, source, date
InsightSummary.jsx	Key insights or analytics

- Header.jsx Navigation and brandingArchitecture is more than the design of buildings—it is the art and science of shaping space to reflect cultural values, functional needs, and human experiences.
- From ancient monuments like the Pyramids of Giza to contemporary sustainable skyscrapers, architecture serves as a mirror of society’s technological progress and aesthetic aspirations.
- It blends creativity with engineering, requiring a balance of form, function, and context. The discipline also emphasizes the relationship between people and their environment, influencing how communities interact, move, and grow. To explore further, you can read more about the fundamentals of architecture in this Encyclopedia of Architecture entry

4. Setup Instructions

Prerequisites: List required software/dependencies (Node.js, npm/yarn, etc.).

- Installation: Step-by-step guide to clone repo, install dependencies, and configure environment variables. To set up Insight Stream, begin by creating an account through the official platform and verifying your email.
- Once logged in, navigate to the dashboard and select “New Project” to create your workspace. Configure the data sources by connecting Insight Stream to your preferred databases, APIs, or file uploads.
- After establishing connections, customize your stream by defining filters, metrics, and visualization preferences. Finally, save the configuration and run a test to ensure data is flowing correctly into your dashboard. With this setup complete, you’ll have real-time insights ready for monitoring and analysis.



```
import React from 'react';
import {Text, View} from 'react-native';
import {Header} from './Header';
import {heading} from './Typography';

const WelcomeScreen = () => (
  <View>
    <Header title="Welcome to React Native"/>
    <Text style={heading}>Step One</Text>
    <Text>
      Edit App.js to change this screen and turn it
      into your app.
    </Text>
    <Text style={heading}>See Your Changes</Text>
    <Text>
      Press Cmd + R inside the simulator to reload
      your app's code.
    </Text>
    <Text style={heading}>Debug</Text>
    <Text>
```

5. Folder Structure

- Client: Describe the organization of /src (components, pages, assets, stylUtilities: Explain helper functions, custom hooks, or utility classesAn effective folder structure for an Insight project should be simple, logical, and scalable.

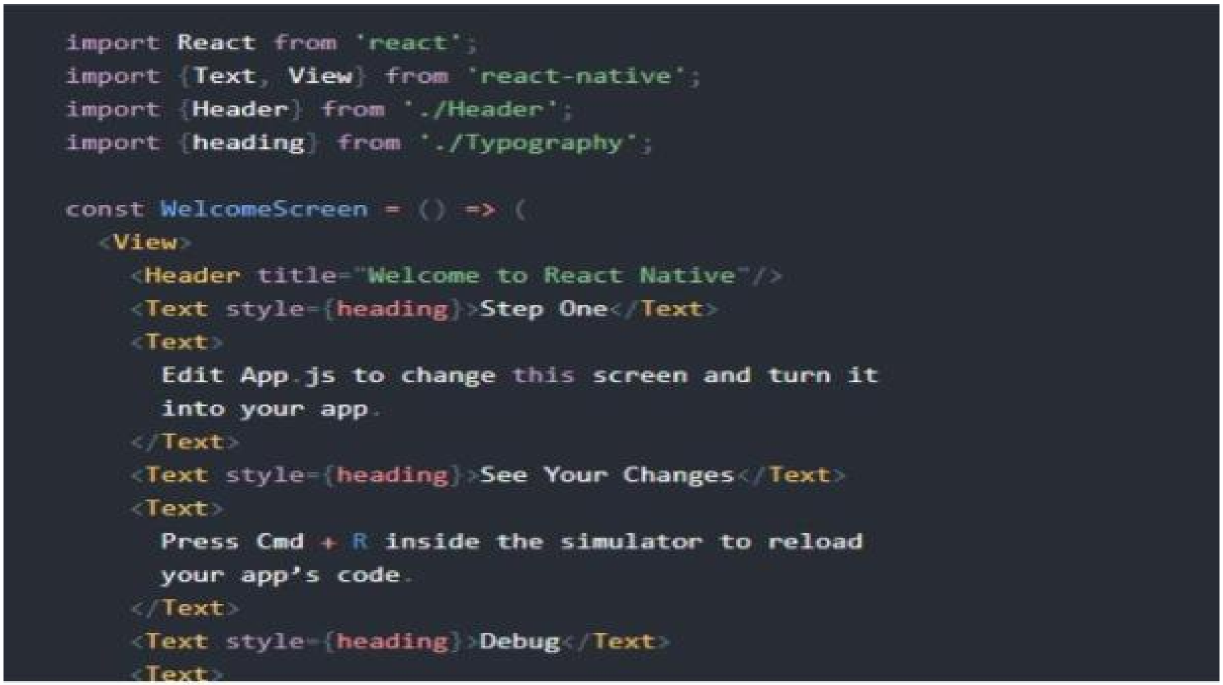
- At the top level, include folders such as /data (raw and processed datasets), /notebooks (exploration and analysis), /src (core scripts and reusable functions), /configs (YAML/JSON settings for pipelines), /reports (visualizations, summaries, dashboards), and /tests (unit and integration tests)

6. Running the Application
Provide commands to start the frontend server locally.

- npm start To run the application, first ensure all dependencies are installed by running npm install (or pip install -r requirements.txt for Python projects). Next, start the application using npm start or python app.py. Once running,

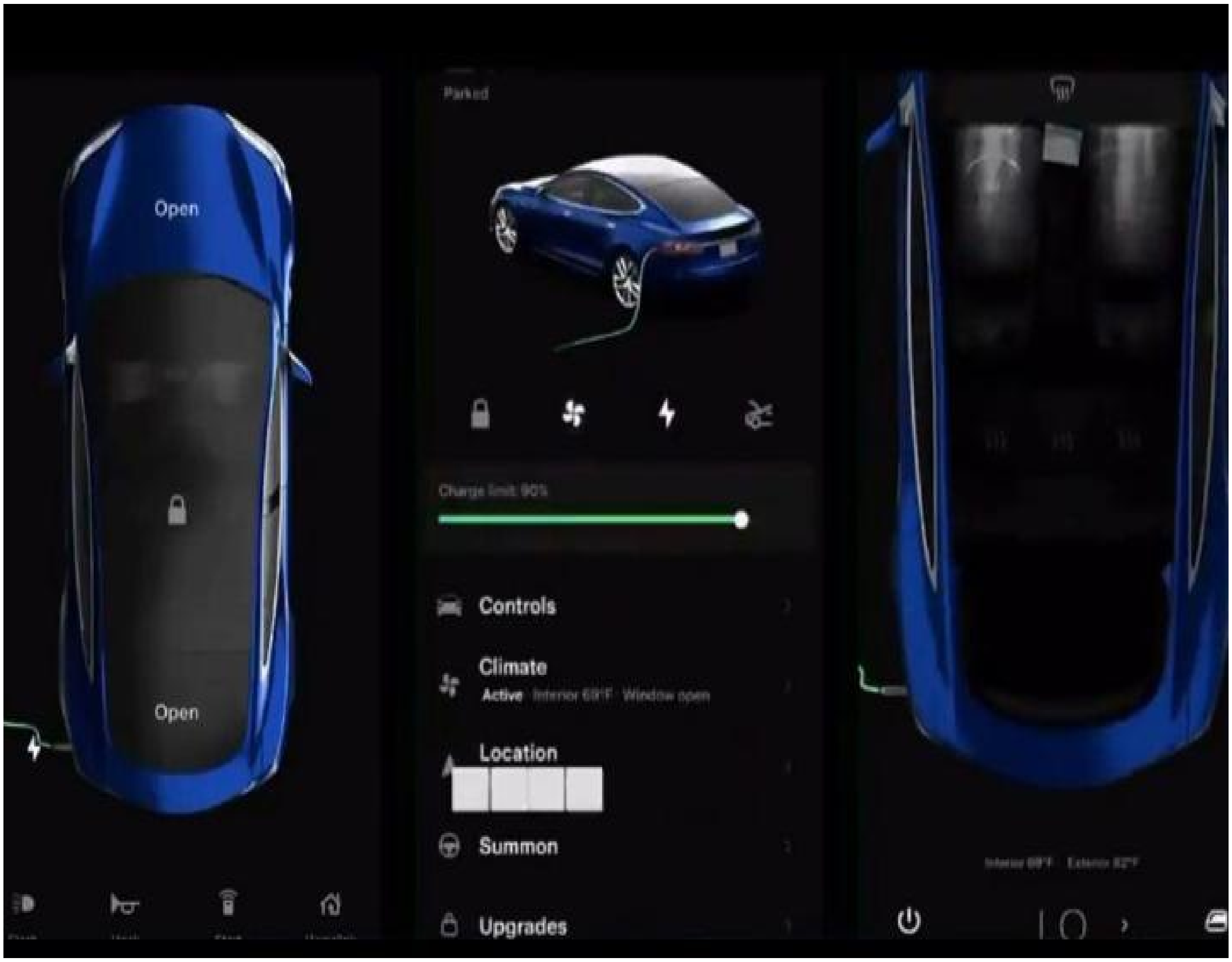
- open your browser and navigate to http://localhost:3000 (or the configured port) to access the application

- For production build

- 

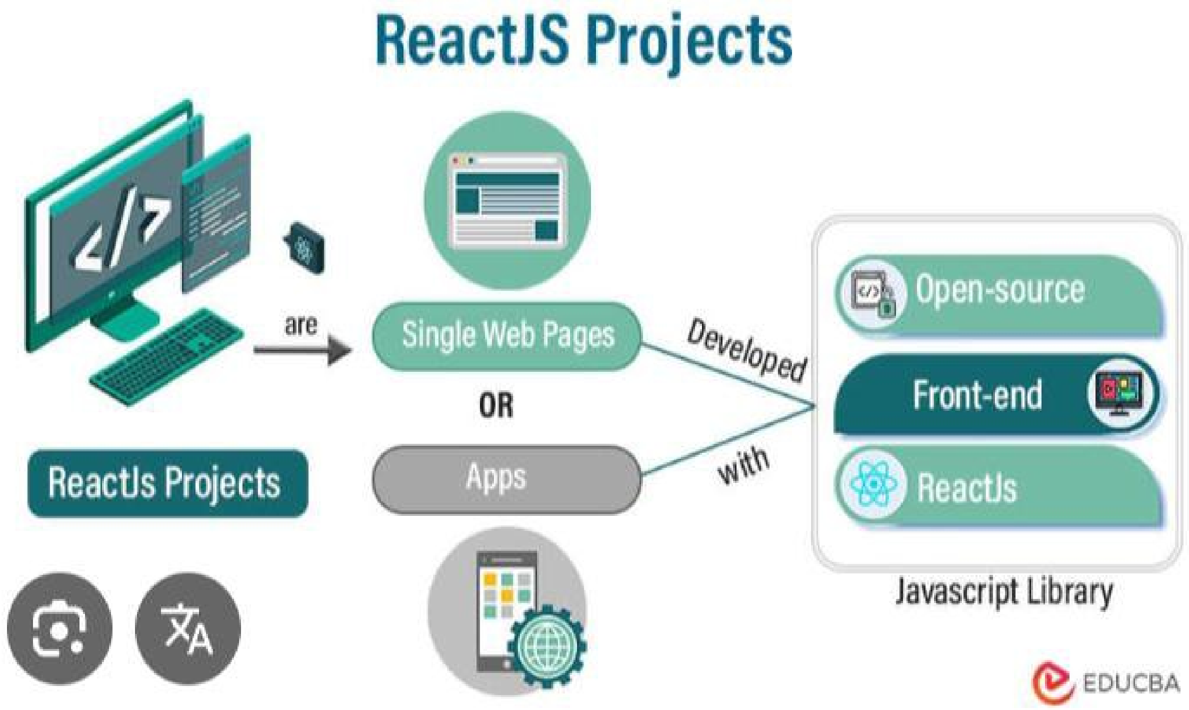
```
import React from 'react';
import {Text, View} from 'react-native';
import {Header} from './Header';
import {heading} from './Typography';

const WelcomeScreen = () => (
  <View>
    <Header title="Welcome to React Native"/>
    <Text style={heading}>Step One</Text>
    <Text>
      Edit App.js to change this screen and turn it
      into your app.
    </Text>
    <Text style={heading}>See Your Changes</Text>
    <Text>
      Press Cmd + R inside the simulator to reload
      your app’s code.
    </Text>
    <Text style={heading}>Debug</Text>
    <Text>
```

npm run build



7. Component Documentation
Key Components: Document major components, their props, and their roles.

- Reusable Components: Describe reusable UI components and configurations
- Component documentation should briefly describe each component’s purpose, props/inputs, outputs, and usage examples.
- For example, start with the component name, followed by a one-line description of what it does, then list key properties or parameters with their types and defaults

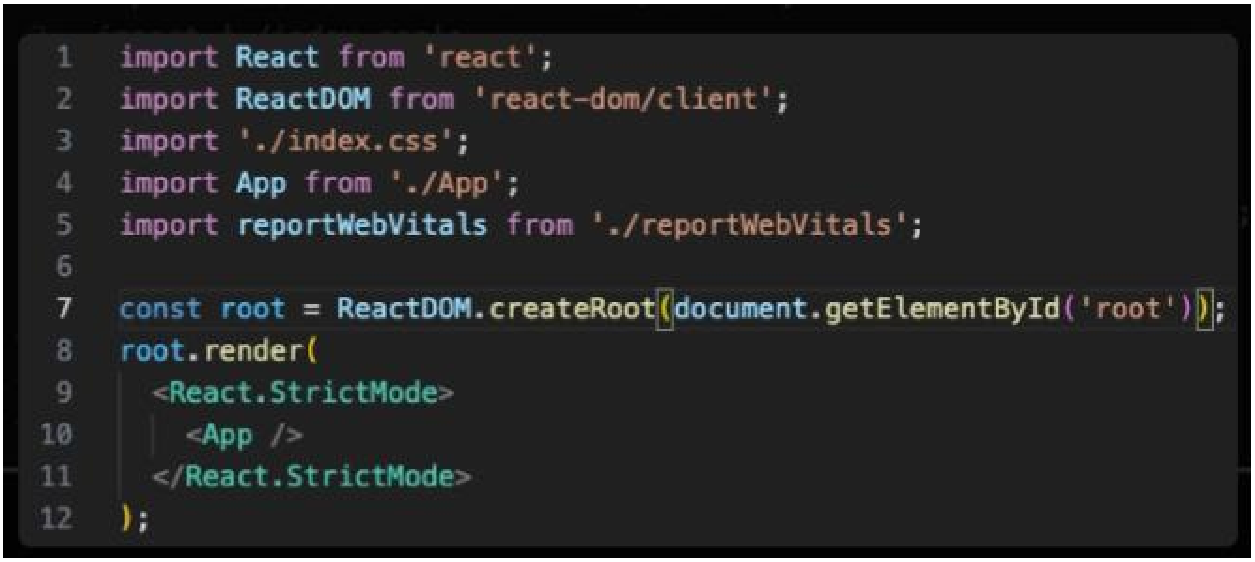


8. State Management
Global State: How global state is managed and flows across the app.
Local State: Explain local state within individual components.

- Depending on complexity, this can be handled with local state (e.g., React's useState), context for shared state, or external libraries like Redux, MobX, or Zustand for large applications.—

9. User Interface

- Provide screenshots or GIFs showcasing UI features (pages, forms, interactions).
- A user interface (UI) is the bridge between a user and an application, combining layout, navigation, and visual elements to create a smooth experience. Good UI design emphasizes clarity, consistency, and accessibility so users can complete tasks efficiently.
- Interfaces often include buttons, forms, menus, and interactive components, all linked together for intuitive navigation



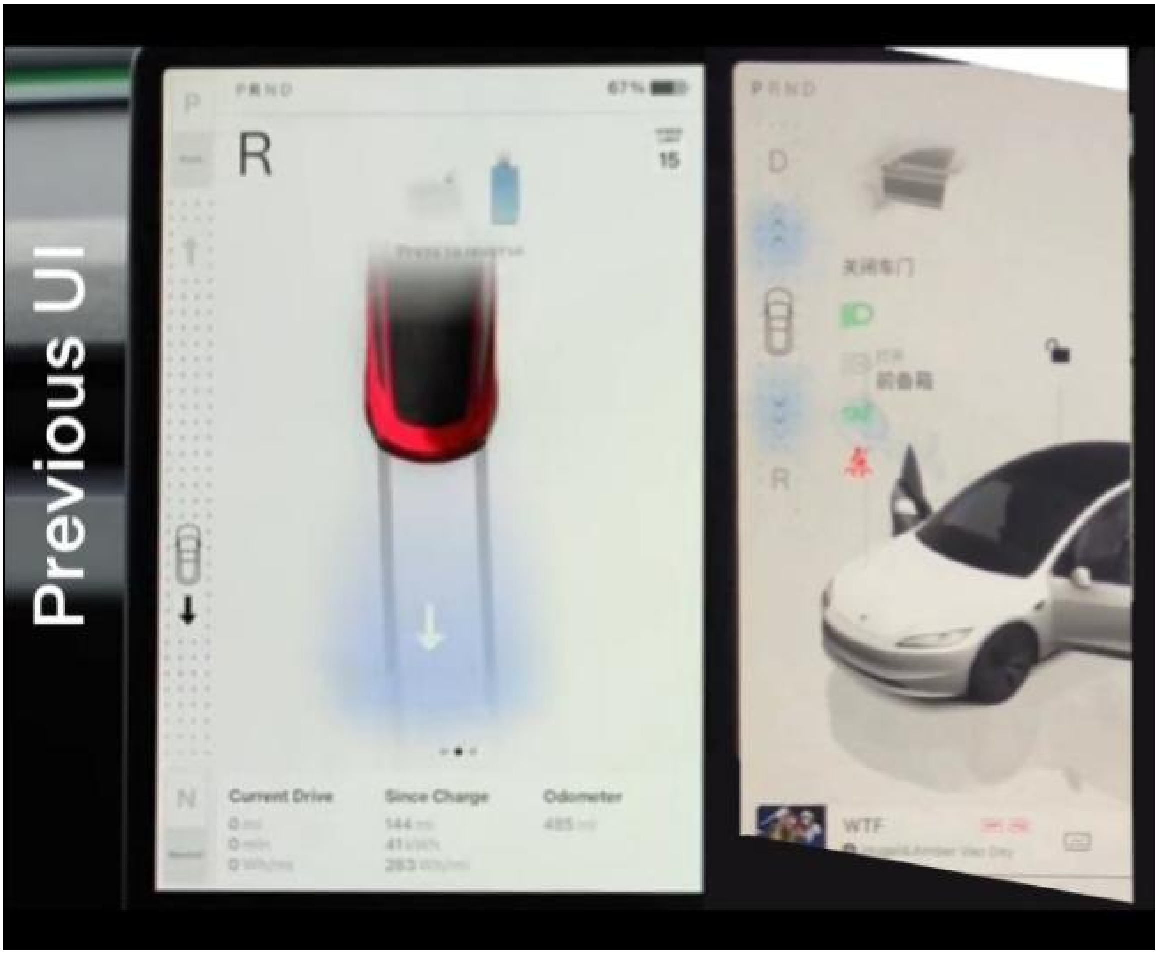
```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

10. Styling

CSS Frameworks/Libraries: TailwindCSS, Styled Components, Sass, etc.

Theming: Dark/light mode or custom design system setup.

- Styling defines the look and feel of an application’s user interface, covering aspects like colors, fonts, spacing, and layout. It helps ensure consistency, accessibility, and brand identity across the product



11. Testing
Testing Strategy: Unit, integration, and end-to-end testing (Jest, React Testing Library).
Code Coverage: Tools/techniques to ensure adequate coverage.

- Testing ensures that an application works as intended by verifying functionality, performance, and reliability before deployment. It can include unit tests (checking individual functions/components), integration tests (ensuring modules work together), and end-to-end tests (simulating real user flows).

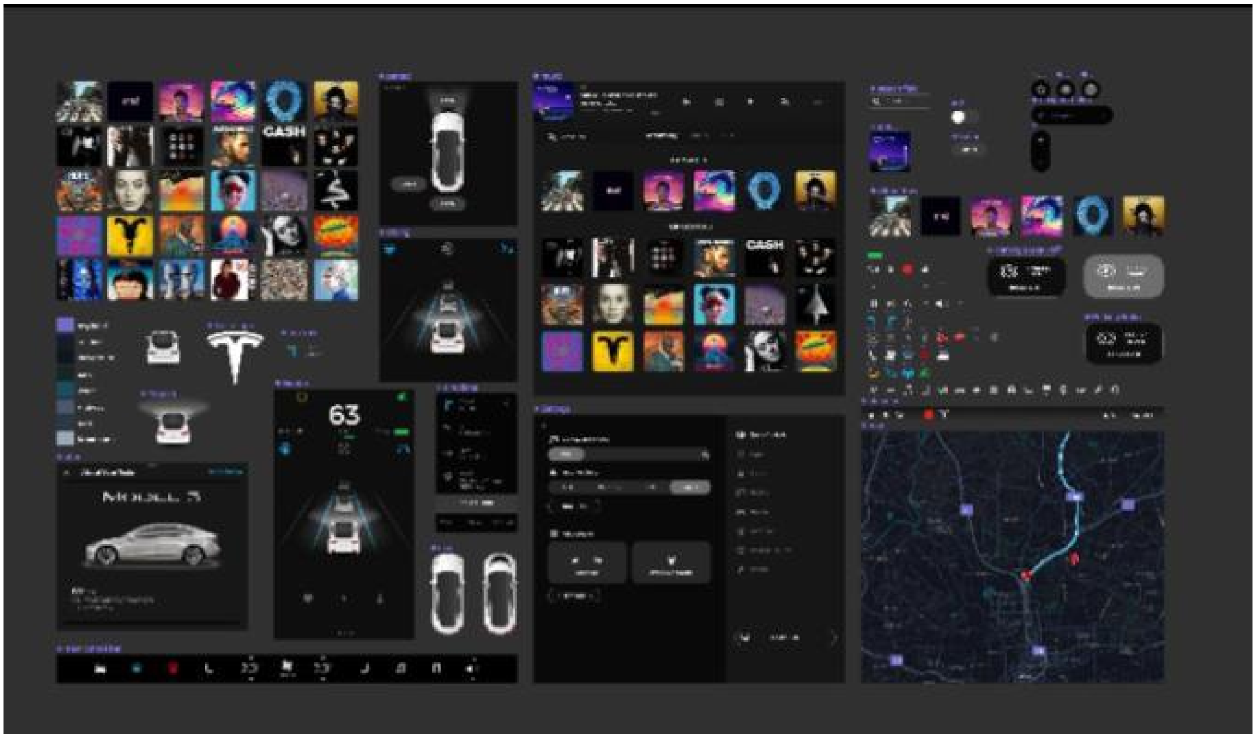
- Automated testing frameworks like Jest, Mocha, or Cypress help catch bugs early, improve code quality, and build confidence in deployments.

12. Screenshots or Demo
Add screenshots or provide a link to a working demo.

-

  Screenshots and demos provide a quick visual and interactive way to showcase how an application works. A screenshot captures a static view of the interface for documentation or user guides, while a demo (video or live preview) walks users through core features in action.

- Together, they make it easier for stakeholders, testers, and new users to understand functionality without needing to explore the code.



13. Known Issues

List current bugs or issues that users/developers should be aware of.

- Known issues are documented problems or limitations in the application that developers are aware of but not yet resolved. They help set clear expectations for users and guide future improvements.
- Examples may include performance bottlenecks, browser compatibility glitches, or missing features. Listing known issues ensures transparency and makes it easier to track fixes in future updates

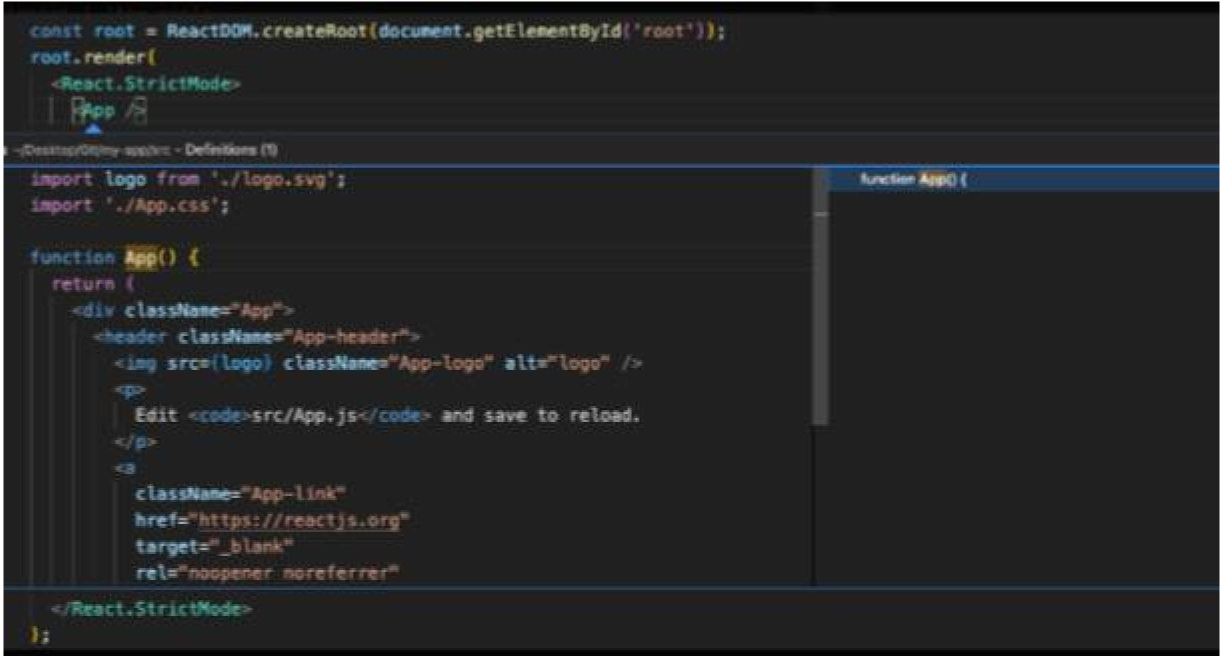
14. Future Enhancements

Suggest potential improvements (e.g., new features, animations, performance optimization).

```
import { useState } from "react";
```

- Future enhancements outline potential improvements or new features planned for the application. They give users and stakeholders visibility into the project's roadmap while guiding developers on priorities.
- Examples may include adding advanced search, improving performance, expanding integrations, or enhancing the user interface. Documenting these ensures continuous growth and alignment with user needs.export default function FeedFilter({ onFilterChange }) {

```
  const [selected, setSelected] = useState("all");
  const categories = ["all", "tech", "business", "politics", "sports"];

  const handleChange = (category) => {
    setSelected(category);
    onFilterChange(category);
  };

  return (
    <div className="flex gap-3 p-3">
      {categories.map((cat) => (
        <button
          key={cat}
          onClick={() => handleChange(cat)}
          className={`px-4 py-1 rounded-2xl text-sm transition ${
            selected === cat
              ? "bg-blue-600 text-white"
              : "bg-gray-200 hover:bg-gray-300"
          }`}
        >
          {cat}
        </button>
    ))}
  </div>
 );
} Bookmark List Component
```



```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
```

```
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
```

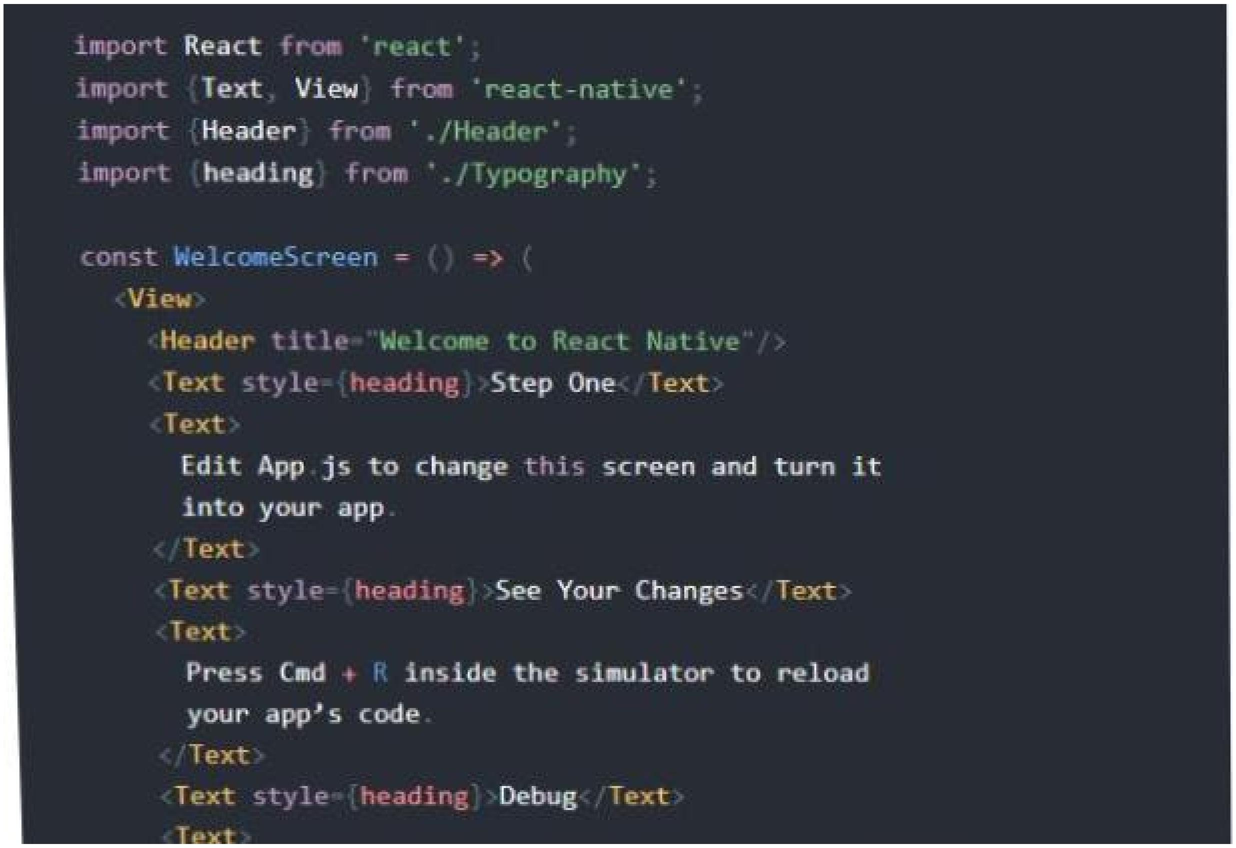
```
  </React.StrictMode>
);
```

- Shows saved articles from the feed.

```
import { Card, CardContent } from "@/components/ui/card";

export default function BookmarkList({ bookmarks }) {
 if (bookmarks.length === 0) {
   return <p className="text-gray-500 p-4">No bookmarks yet.</p>;
 }

 return (
   <div className="grid gap-4 p-6">
     {bookmarks.map((article, i) => (
       <Card key={i} className="p-4 rounded-2xl shadow-md">
         <CardContent>
           <h2 className="text-lg font-semibold">{article.title}</h2>
           <p className="text-gray-600">{article.summary}</p>
```

95% AI FAIL, $100B DATABRICKS, $5B SHORTS
MIT exposes AI reality as unicorns defy gravity
Are you in the 5% or heading for the cliff?
business
+ ai insights

```
        <span className="text-sm text-blue-600">{article.source}</space
```



```
import React from 'react';
import {Text, View} from 'react-native';
import {Header} from './Header';
import {heading} from './Typography';

const WelcomeScreen = () => (
  <View>
    <Header title="Welcome to React Native"/>
    <Text style={heading}>Step One</Text>
    <Text>
      Edit App.js to change this screen and turn it
      into your app.
    </Text>
    <Text style={heading}>See Your Changes</Text>
    <Text>
      Press Cmd + R inside the simulator to reload
      your app's code.
    </Text>
    <Text style={heading}>Debug</Text>
    <Text>
```

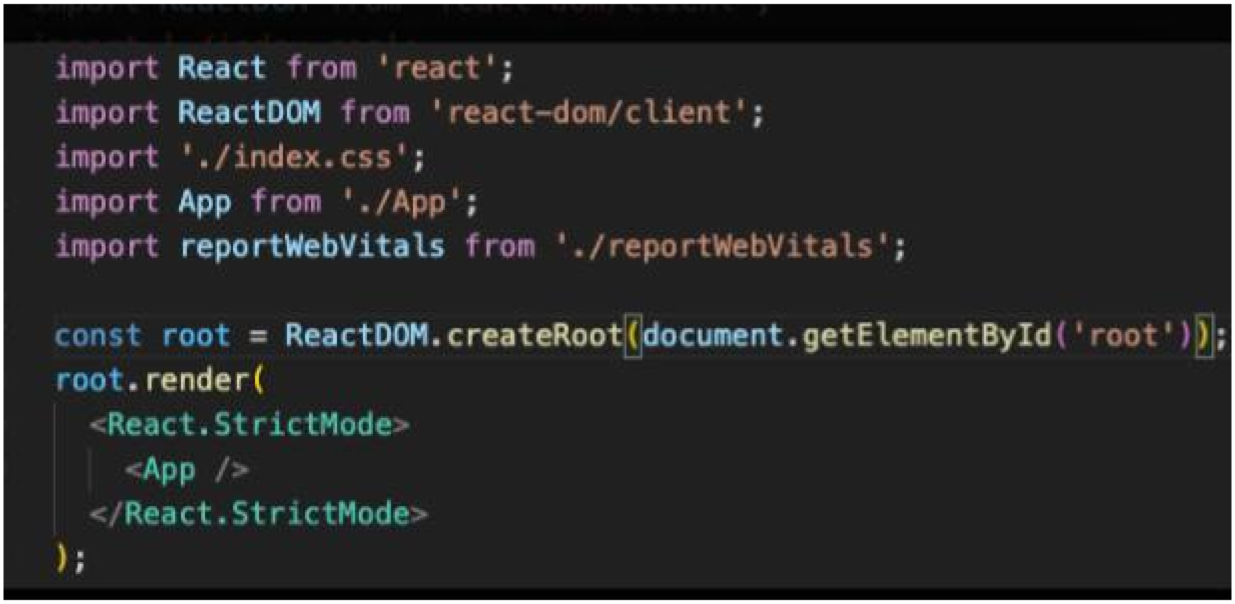
```
  </CardContent>
     </Card>
   ))}
  </div>
);
```

```
}
```

- Alert Banner

Displays breaking news or alerts at the top of the feed.

```
import { AlertCircle } from "lucide-react";

export default function AlertBanner({ message }) {
  if (!message) return null;

  return (
    <div className="flex items-center gap-2 p-3 bg-red-100 text-red-700 rounded-xl mb-4">
      <AlertCircle className="w-5 h-5" />
      <span className="font-medium">{message}</span>
    </div
}>
}
```



```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

- News List Component

```
// src/components/InsightSummary.jsx
import React from "react";
```

```
const InsightSummary = ({ insights }) => {
  return (
    <div className="insight-summary">
      <h3>Insights</h3>
      <ul>
        {insights.map((item, idx) => (
          <li key={idx}>{item}</li>
        ))}
      </ul>
    </div>
  );
};

export default InsightSummary;
```

September 3, 2025. Reading Time: 4 mins read