**Recipe Finder Documentation**

**Introduction:**

The Recipe Finder is a React-based web application that allows users to search for recipes and view detailed information about them. This documentation provides a comprehensive guide to the project's structure, implementation, and usage.

**Project Summary:**

The Recipe Finder application is built using:

- React.js for the frontend

- Vite as the build tool

- Bootstrap for styling

- React Icons for visual elements

**Key features include:**

- Search functionality with auto-suggestions

- Recipe details display

- Responsive design

- Background image for visual appeal

**Project Aim:**

The main objectives of this project are:

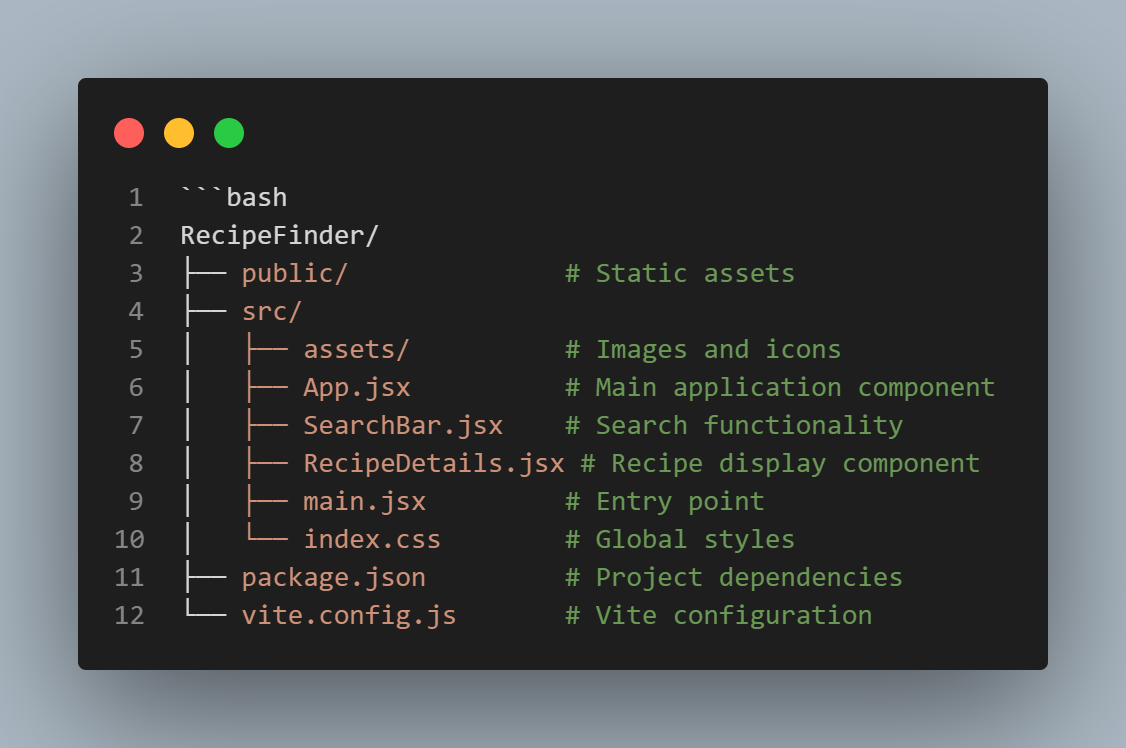
1. To create an intuitive interface for recipe searching

2. To demonstrate React component-based architecture

3. To implement efficient state management

4. To showcase modern web development practices

**Project Structure:**



**Implementation with Code Explanations:**

1. **Main Application (App.jsx)**

A screen shot of a computer program

AI-generated content may be incorrect.

**- Explanation:**

- Uses React hooks (useState, useEffect) for state management

- Manages three main states: search query, recipes list, and selected recipe

- useEffect hook triggers recipe fetching when query changes

- Renders SearchBar and RecipeDetails components conditionally

1. **Search Component (SearchBar.jsx)**



**- Explanation:**

- Receives props from parent component for state management

- Manages dropdown visibility state

- Handles input changes and updates query state

- Renders dropdown with recipe suggestions

- Implements recipe selection functionality

1. **Recipe Details Component (RecipeDetails.jsx)**
2. import React from "react";
3. import { IoStarSharp } from "react-icons/io5";
4. const RecipeDetails = ({ recipe }) => {
5. return (
6. <div className="recipe-card">
7. <div className="img-container">
8. <img src={recipe.image} alt="recipe-image" className="recipe-image" />
9. </div>
11. <div className="details-container">
12. <h2 className="title">{recipe.name}</h2>
13. <div className="details">
14. <span className="fw-bold">Ingredients:</span>
15. <ul className="ingredients">
17. {recipe.ingredients.map((ingredient, index) => (
18. <li key={index}>{ingredient}</li>
19. ))}
20. </ul>
22. <p className="instructions"><span className="fw-bold">Instructions:</span>{recipe.instructions}</p>
23. <p className="preptime"><span className="fw-bold">Prepration Time:</span> {recipe.prepTimeMinutes} minutes</p>
24. <p className="cooktime"><span className="fw-bold">Cooking Time:</span> {recipe.cookTimeMinutes} minutes</p>
25. </div>
26. <div className="additional-info">
27. <p className="servings"><span className="fw-bold"> No of Servings: </span>  {recipe.servings}</p>
28. <p className="calories"><span className="fw-bold">Calories per Serving: </span> {recipe.caloriesPerServing}</p>
29. <div className="rating">
30. <span className="fw-bold">Rating: </span> {'⭐'.repeat(Math.round(recipe.rating))}{'⭐'.repeat(5 - Math.round(recipe.rating))}
31. </div>
32. <p className="review"><span className="fw-bold">Reviews: </span>{recipe.reviewCount}</p>
33. </div>
34. </div>
35. </div>
36. );
37. };
38. export default RecipeDetails;

- **Explanation:**

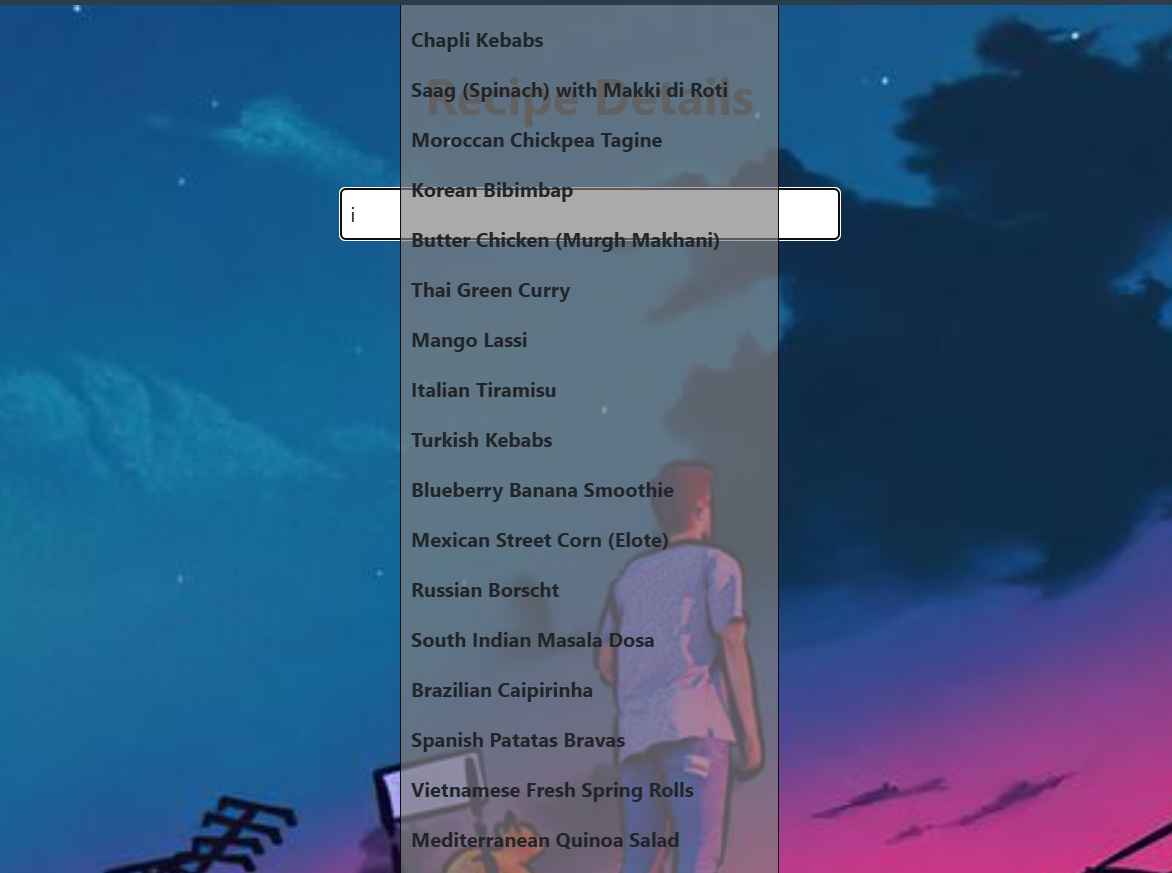
- Receives selected recipe as prop

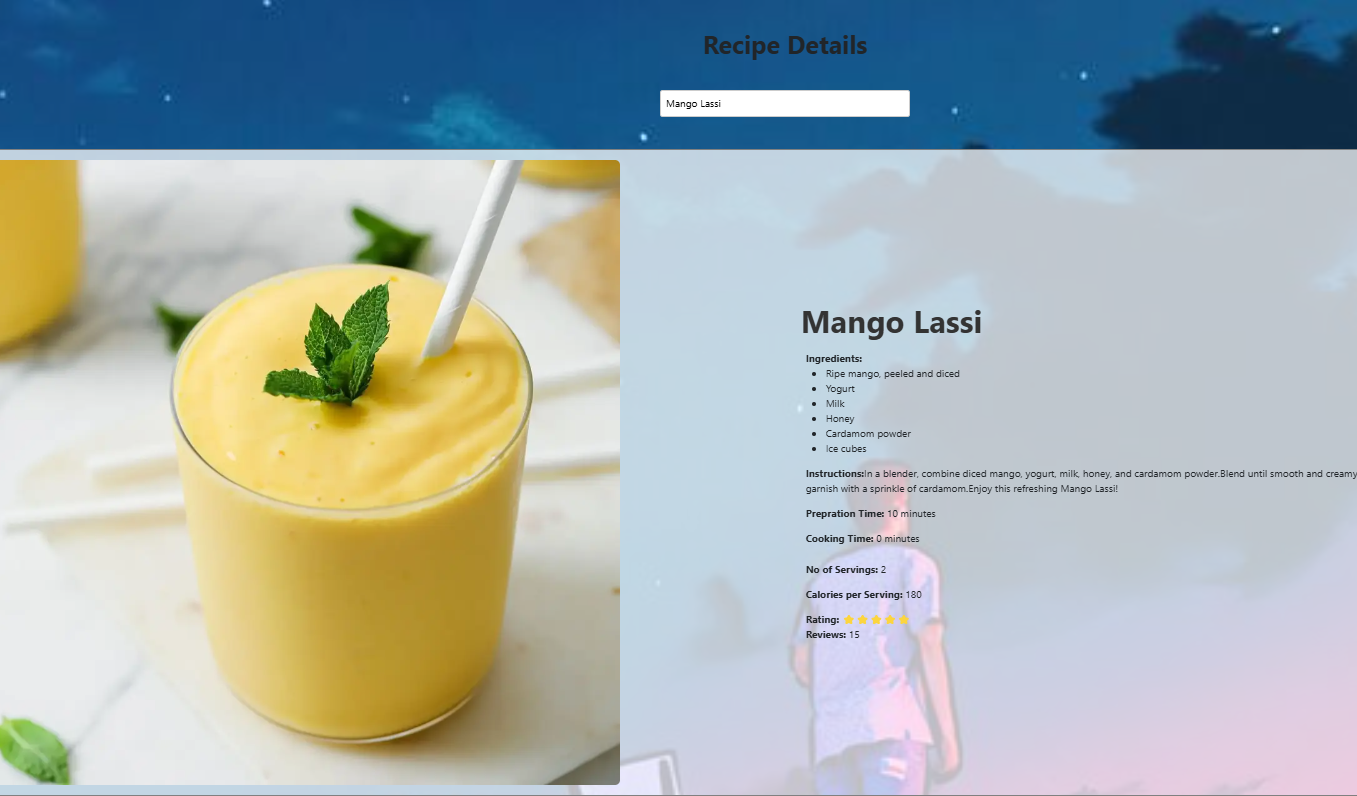
- Displays recipe image, name, description, and rating

- Uses React Icons for star rating display

- Implements responsive card layout

**Expected Output:**

****

****

1. Search bar with dropdown suggestions

2. Recipe cards with detailed information

3. Responsive layout for different screen sizes

**Conclusion:**

The Recipe Finder application demonstrates modern web development practices using React and Vite. It showcases component-based architecture, state management, and responsive design. This project serves as a foundation for building more complex recipe-related applications.