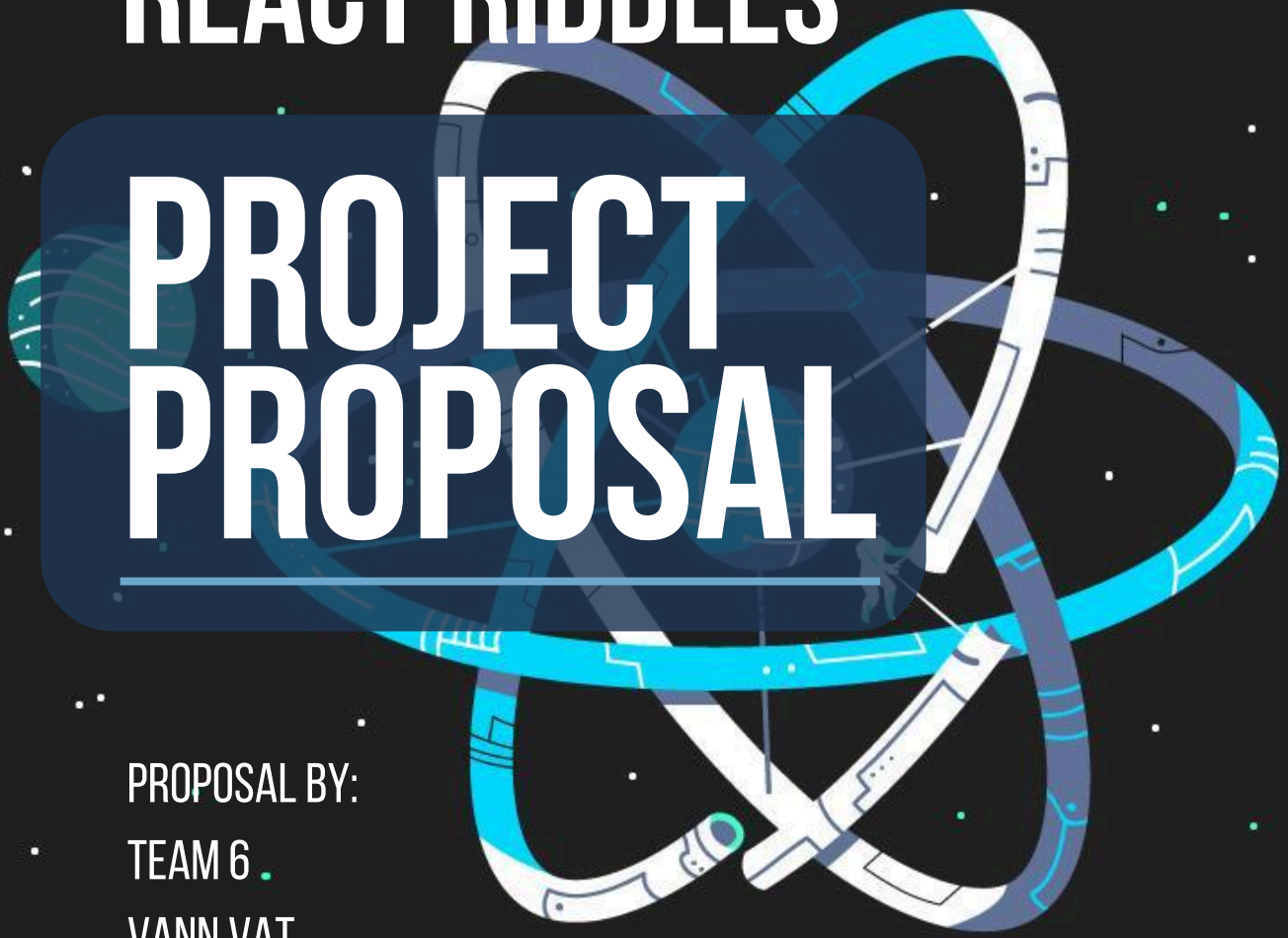
CADT
IDT វិទ្យាស្ថានបច្ចេកវិទ្យាឌីជីថល
Institute of Digital Technology

2025

REACT RIDDLES

PROJECT PROPOSAL



PROPOSAL BY:

TEAM 6

VANN VAT

CHIM PANHAPRASITH

PHAN PANHAWORND

DEPARTMENT:

Computer Science

Data Science

DIRECTED BY: LECT. SOK PISETH

Table of Contents

1. Introduction

Our React Quiz Game is called "**React Riddles**" is a fun and interactive way to learn React. It is designed to help users practice important React concepts step by step. The game has nine levels, and each level focuses on a specific topic, such as props, state, hooks, conditional rendering, arrow function, and event handling.

In each level, you are given a challenge to solve by typing the correct React code. If your answer is correct, you will see a success message, and you can move on to the next level by pressing "Enter." and if you want to navigate to the different level, you can use arrow icons on the screen.

This game is perfect for beginners who want to learn React or for developers who want to refresh their skills.

2. Understanding Your Need

Based on our discussion from our team, we need to complete on:

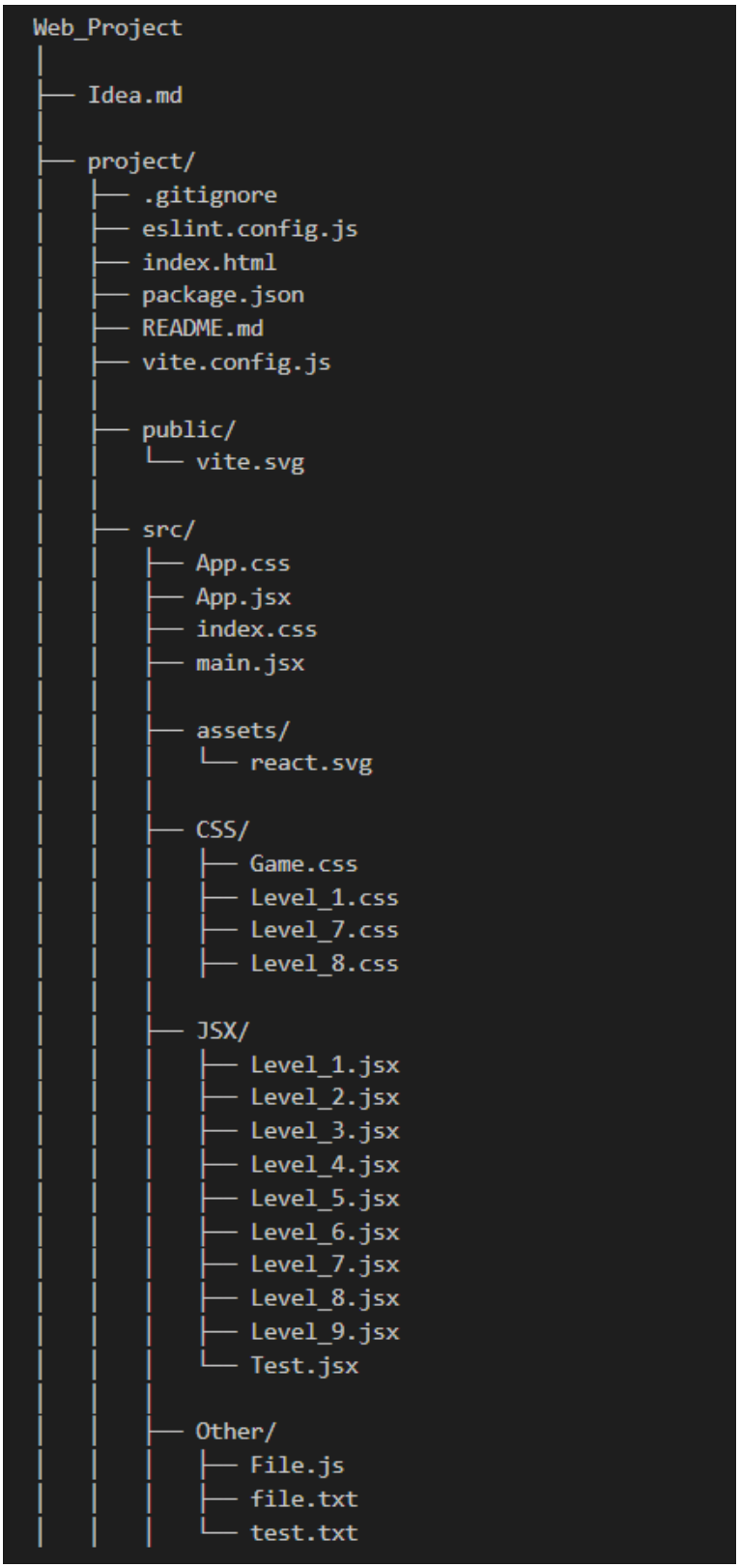
- **Responsive Design**: The game should work well on all devices, including desktops, tablets, and mobile phones.
- **Interactive Levels**: Each level should present a React-related challenge, allowing users to type their answers and get real-time feedback.
- **Progression System:** Users should automatically move to the next level after pressing enter once users are completing the current one.
- **Educational Content:** Each level should explain the React concept being tested, helping users learn as they play.
- **Dynamic Feedback:** The game should provide instant feedback for correct and incorrect answers, guiding users to improve.
- **Scalable Architecture:** The game should be easy to update and expand with new levels in the future.
- **Level Navigation:** Users should also have the option to navigate to different levels directly, giving them flexibility to revisit or skip levels as needed.

3. Proposed Solutions

proposed solutions aim to create a user-friendly, educational, and scalable React Quiz Game:

1. **Responsive Design:**
   - Use responsive CSS techniques (e.g., media queries) to ensure the game works seamlessly on desktops, tablets, and mobile phones.
2. **Interactive Levels:**
   - Design each level to present a React-related challenge with an input field for users to type their answers.
3. **Progression System:**
   - Create a system that tracks the user's current level and unlocks the next level only after the correct answer is submitted.
4. **Educational Content:**
   - Include a brief explanation of the React concept being tested at the start of each level.
   - Provide examples or hints to help users understand the topic and solve the challenge.
5. **Dynamic Feedback:**
   - Implement a feedback system that displays messages for correct, incorrect.
6. **Scalable Architecture:**
   - Use a modular structure for levels, with each level having its own component.
   - Ensure the architecture allows for easy addition of new levels without affecting existing functionality.
7. **Level Navigation:**
   - Add a navigation button to allow users to jump to specific levels directly.

4. Code Structure



5. Methodology

- Planning and set up: Define features and plan the project. Using tools such as Visual Studio Code.
- Communication and Collaboration tools: Github, Telegram, Discrd
- Library used: React
- Styling the website: CSS
- Vite: Lightweight build tool for web development, optimizing performance.

6. Project Timeline

- **Week 1:**
  - Finding topic, idea for the project
  - Define all the level of the quiz
  - Design each level
- **Week 2:**
  - Create react project, write each level code into its file
  - Design style of each level
  -
- **Week 3:**
  - Testing the code, fixing error

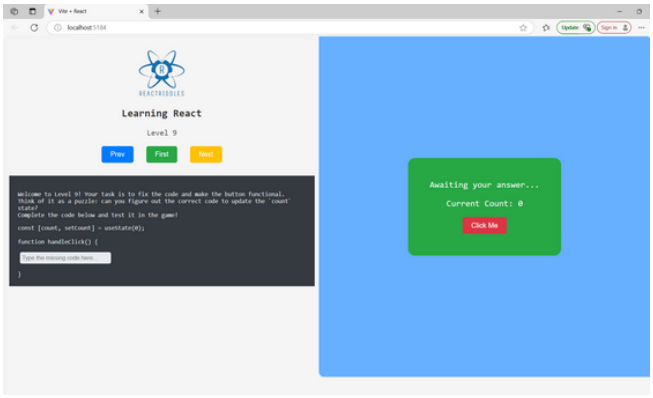
7. Expected Outcome

The website will be able to:

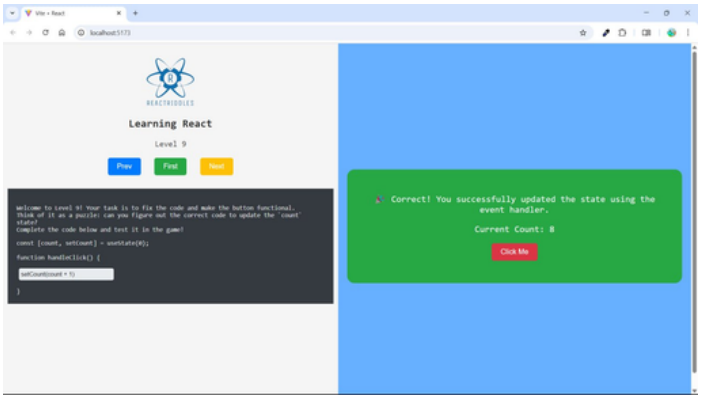
- Interactive Learning:
  - Users progress through levels by solving React-based challenges.
  - Each level introduces a new React concept, such as JSX, props, state, event handling, conditional rendering, and forms.

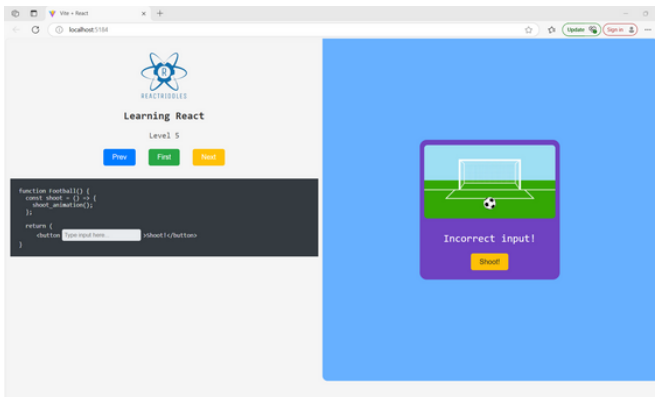
- Visual Feedback:
  - Users receive real-time feedback on their input, helping them learn React concepts effectively.
  - Success messages and animations provide a sense of accomplishment.
- Progression System:
  - Users start at Level 1 and progress to the next level upon completing the current one.
  - After you complete the final level which is (Level 9), it will display a "Thank You for Playing!" message with a celebratory animation.

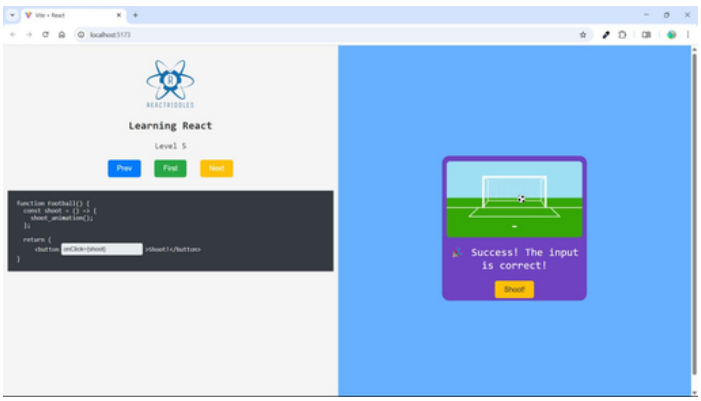
Before Completing Code

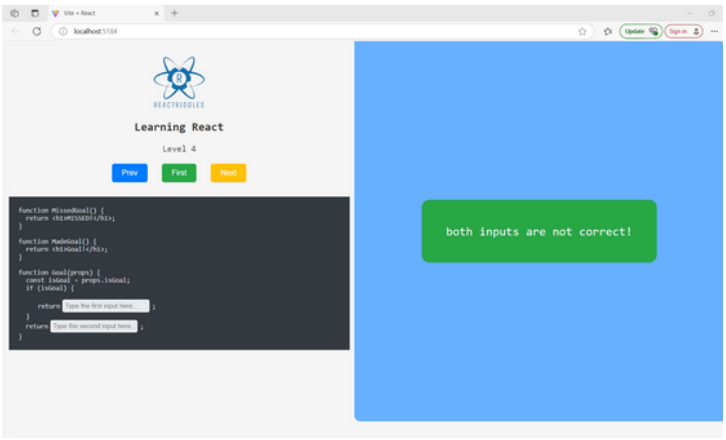


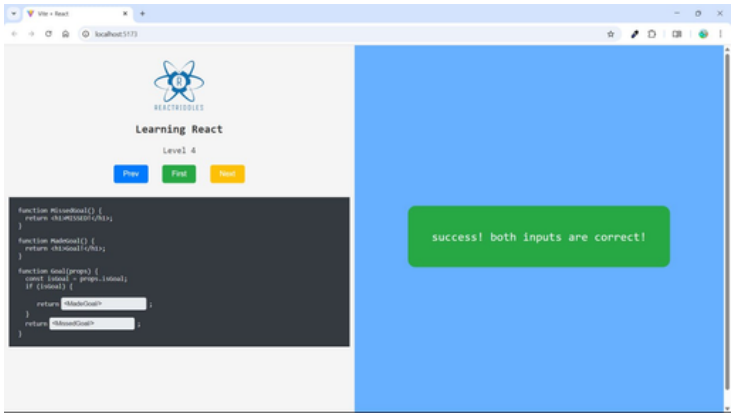
Before Completing Code











8. References

+ W3Schools. (n.d.). W3Schools online web tutorials. from

https://www.w3schools.com/

+ Codepip. (n.d.). Flexbox Froggy - A game for learning CSS flexbox. from

https://flexboxfroggy.com/