Универзитет „Св. Кирил и Методиј“ во Скопје
ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ
И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

GAMESTOP

Documentation

Mentor

Boban Joksimoski

Authors

Dejan Simonovski 211039

Angel Risteski 211095

Skopje, 2024

Contents

Introduction

GameStop VueJS app, a comprehensive project developed as part of the curriculum for the subject 'Advanced Web Design' at FCSE Skopje. This initiative represents the culmination of our efforts to explore and apply what we have learned.

This documentation provides a comprehensive overview of the GameStop VueJS app, covering its technologies, features, components, etc. It is our hope that this project not only fulfills the requirements of our academic curriculum but also serves as a source of inspiration for fellow enthusiasts and aspiring web developers.

Overview

At its core, the GameStop app features four popular games: Snake, TicTacToe, Memory Game, and Hangman. Each game is meticulously crafted to provide engaging gameplay experiences while leveraging the power and flexibility of VueJS, a progressive JavaScript framework for building user interfaces.

Through the seamless integration of classic games, high score tracking combined with user login system, GameStop offers a compelling and immersive gaming platform for users of all ages. Whether seeking nostalgic thrills or competitive challenges, players can find endless entertainment within this dynamic web application.

Technologies Used

- **Vue.js**: Vue.js is a progressive JavaScript framework for building user interfaces. It provides a flexible and efficient way to develop single-page applications, making it ideal for creating dynamic gaming experiences.
- **Vue Router**: Vue Router is the official router for Vue.js. It enables navigation between different views of the application, facilitating a smooth and intuitive user experience.
- **HTML5**: HTML5 provides the structure and semantics for the GameStop VueJS app, allowing for the creation of interactive web pages and games.
- **CSS3**: CSS3 is used for styling and layout of the app, ensuring a visually appealing and consistent user interface across different devices and screen sizes.
- **JavaScript**: JavaScript is the programming language used to add interactivity and functionality to the app, including game logic, user interactions, and dynamic content generation.

Features

- **High Score System**: Users can compete for the top spot on the leaderboard by achieving high scores in each game. The high score system adds a competitive element to the gaming experience, motivating players to improve their skills and surpass their friends' scores.

- **User Authentication**: The app features a secure authentication mechanism that allows users to create accounts, log in, and personalize their gaming experience. User authentication ensures data privacy and security while enabling personalized features such as saving game progress and tracking achievements.

- **Responsive Design**: The app is built using responsive design principles, ensuring compatibility with a variety of devices and screen sizes. Whether users access the app on a desktop, tablet, or smartphone, they can enjoy a seamless and optimized gaming experience.

- **Intuitive User Interface**: The app boasts an intuitive user interface designed for ease of navigation and interaction. Clear instructions, intuitive controls, and visually appealing graphics enhance the user experience and make it easy for players to jump into their favorite games.

Setup

Clone the repository: git clone https://github.com/dejan-simonovski/GameStop.git

1. Navigate to the project directory:

   ```
   cd GameStop
   ```

2. Install dependencies:

   ```
   npm install
   ```

3. Launch the app:

   ```
   npm run serve
   ```

4. Open http://localhost:8080/ in your browser to view the app.

Components

- **Home View**: Upon launching the app, you'll be directed to the Home View. From here, you can explore the available games and access other sections of the app. In case you are not logged in, you will be redirected to the login view.
- **Game Selection**: In the Home View, you'll find options to select different games available in the app. Click on the game to proceed to the respective game view.
- **Game Views**: Once you select a game, you'll be taken to its dedicated Game View. Here, you'll find the game interface along with any additional components specific to that game.
- **Game Components**: Within each Game View, you'll encounter the corresponding Game Component, responsible for handling the game logic, user interactions, and rendering the game interface.
- **Score Component**: The Score Component is integrated into every game, providing real-time feedback on the player's performance.
- **Snake Controls**: If you choose to play Snake, you'll have the option to adjust game settings using the Snake Controls Component. This component allows you to change the game speed and toggle the border between solid and non-solid states.
- **Login/Register**: To access the games and participate in the high score system, you must use the Login/Register Component. This component enables users to create an account, log in, or register as a new user.

Overcoming Obstacles in Development

Throughout the development of the GameStop VueJS app, we encountered various challenges that tested our problem-solving skills and ingenuity. Here's a glimpse into some of the hurdles we faced and the solutions we implemented to overcome them:

- **Snake Game Multi-Spawn Error**: One significant issue arose in the Snake game, where multiple snakes would spawn if users navigated back to the home page and returned to the game. To resolve this, we implemented a solution to reset the game state when the Snake component unmounted in Vue, ensuring a clean start each time the game was played.

- **Snake Disappearing from Game Board**: One obstacle we faced was the issue of the snake looping through to the other side of the border. Through careful analysis and experimentation, we modified the movement functions to ensure that the snake remained within the boundaries of the game board.

- **Memory Game Alignment**: Among these hurdles was the issue of centering the game board, as all div elements were initially inline, causing alignment difficulties. Additionally, because we enhanced the css due to this bug, we decided to go a step further and expand the user experience by incorporating animations to highlight guessed blocks and correct matches.

- **Hangman Game Logic Errors**: In the Hangman game, we faced logic errors related to handling player guesses. Initially, the game did not stop when the number of attempts reached zero, leading to negative attempts. Additionally, the game did not remove an attempt when players guessed an entire word. To address these issues, we added conditional statements to properly manage game flow, ensuring that the game stopped when attempts were depleted and correctly deducted attempts when guessing words.

By identifying and addressing these challenges head-on, we were able to enhance the functionality and stability of the GameStop VueJS app, delivering a seamless and enjoyable gaming experience for our users. These experiences served as valuable learning opportunities, reinforcing the importance of thorough testing, meticulous problem-solving, and effective collaboration in software development endeavors.

Conclusion

As we reflect on the development journey of the GameStop VueJS app, we are proud of the achievements and lessons learned along the way. We hope that this project not only meets the academic requirements of our curriculum but also serves as a source of inspiration for fellow developers and enthusiasts in the web design community.

Thank you for joining us on this journey, and we invite you to explore the GameStop VueJS app, play your favorite games, and experience the thrill of classic gaming in a modern, interactive setting.