

Project Title: Blackjack Card Counting Training Website Student

Name: Robert Nguyen

Email Address: rhnguyen@go.olemiss.edu

Sponsor Name: Dr. Yixin Chen

Organization: University of Mississippi Department of Computer Science

Sponsor Email: yixin@olemiss.edu

Blackjack System Requirements Document

Project Overview

The Blackjack Card Counting Training Website is an educational web application aimed at teaching users the mathematical concepts behind card counting. While often associated with gambling, this project focuses purely on probability, strategy, and mental discipline. Many existing card counting resources are either text-heavy or require specialized software, making them inaccessible to a broad audience. This project seeks to create an intuitive and interactive platform that allows users to learn through hands-on practice, reinforcing theoretical concepts via real-time blackjack simulations. The platform will provide structured learning modules, immediate feedback, and performance tracking to ensure an engaging and effective learning experience.

User Requirements:

The Minimum Viable Product (MVP) will be a functional web application designed to teach users the fundamentals of card counting in blackjack. It will include an interactive simulator where users can practice counting cards in a controlled environment, along with an educational module that explains key strategies and techniques. The platform will present information in a structured, easy-to-follow format, making it accessible to both beginners and those looking to refine their skills. Users will be able to engage with lessons at their own pace, reinforcing their understanding through guided practice. In addition to learning materials, the simulator will provide real-time feedback to help users improve their accuracy and decision-making. A simple and intuitive interface will ensure a seamless learning experience. The MVP will focus on delivering a solid foundation in card counting, with potential future enhancements to expand on advanced strategies and analytics.

User Types:

- **Players** – Individuals who want to learn and practice card counting strategies
- **Admins** – Users who oversee and manage the platform, ensuring a smooth and secure experience

User Stories:

Player:

- 1.) As a player, I want to **create an account** so that I can save my progress, track my learning, and access advanced features of the platform.
- 2.) As a player, I want to **play a simulated blackjack game** so that I can practice card counting techniques in real-time, reinforcing my learning through interactive gameplay.

Administrator:

- 3.) As an admin, I want to **approve and verify new users** so that I can ensure the integrity of the platform and prevent unauthorized access.
- 4.) As an admin, I want to **monitor and moderate the platform** to ensure fair play and prevent users from exploiting the system.

Design Choices:

Frontend

React.js

- Advantages: Component-based structure, efficient state management, widely adopted in modern web development.
- Disadvantages: Learning curve for beginners, requires additional libraries for state management.

Vue.js

- Advantages: Lightweight, easy to learn, great for small to mid-sized applications.
- Disadvantages: Smaller community than React, limited large-scale application support.

Backend

Node.js with Express.js

- Advantages: Fast, non-blocking architecture, well-integrated with frontend frameworks.
- Disadvantages: Can become complex with multiple dependencies, requires careful resource management.

Flask (Python)

- Advantages: Lightweight, simple to implement, great for small to medium-sized applications.

- Disadvantages: Not as scalable as other options, requires additional configurations for large-scale applications.

PHP

- Advantages: Well-supported for web development, especially for applications requiring server-side scripting.
- Disadvantages: Older technology with performance limitations, not as efficient for handling large-scale real-time interactions.

Database

MariaDB (SQL-based)

- Advantages: Reliable, ACID-compliant, great for structured relational data such as user statistics.
- Disadvantages: Requires predefined schemas, less flexible than NoSQL databases for handling unstructured data.

MongoDB (NoSQL)

- Advantages: Flexible schema, scalable for handling large amounts of unstructured data.
- Disadvantages: Not ideal for structured relational data, requires different querying approaches compared to SQL-based databases.

Task Scheduling:

- Week 4-8: Front and Back-end Development (website structure, game logic, user account, database storage)
- Week 9-10: Interactive Features (progress tracking)
- Week 11-12: Testing
- Week 13-14: Final Adjustments
- Week 15: Presentation