

# Blackjack Card Game

---

A modern Blackjack card game built with SvelteKit and TypeScript, using the DeckOfCards API for card management.

## Project Overview

This project implements a fully functional Blackjack game with an intuitive user interface. The game follows standard Blackjack rules, allowing players to hit, stand, double down, and split pairs while playing against a dealer who follows conventional house rules.

## Tech Stack

- **Framework:** SvelteKit
- **Language:** TypeScript
- **API:** [Deck of Cards API](#)
- **Styling:** CSS (no frameworks)

## Code Conventions

### Naming Conventions

- **Branches** type/teamname/ticketid-description (feature/ bugfix/, hotfix/, release/, etc.)
- **Classes:** PascalCase (**P**layer**H**and, **D**ealer**S**core)
- **Variables:** camelCase (**d**eck**s**ervice, **g**ame**s**tore)
- **Functions:** Arrow functions when possible

### Syntax Preferences

- **Strings:** Double quotes ("example") instead of single quotes
- **Statements:** Semicolons required
- **Code Organization:**
  1. Imports
  2. Exports
  3. Definitions
  4. Functions

### Architecture Patterns

- **API Interactions:** Object-based approach with method chaining

```
// Example
DeckApi.deck.shuffle(deckId);
DeckApi.deck.cards.draw(deckId, count);
```

- **State Management:** Class-based Svelte stores

```
// Example
export class GameStore extends writable<GameState> {
  constructor() {
    super(initialState);
  }

  startNewGame = () => {
    // Implementation
  };
}
```

## Project Structure

```
src/
├── lib/
│   ├── services/
│   │   ├── apiname/
│   │   │   ├── ApiName.ts           # Class Based Api Implementation
│   │   │   └── endpointname/
│   │   │       └── EndpointName.ts # Class Based Api Endpoint
│   ├── components/
│   │   ├── Card.svelte             # Card display component
│   │   ├── Hand.svelte             # Hand display component
│   │   ├── Dealer.svelte           # Dealer component
│   │   ├── Player.svelte           # Player component
│   │   ├── Actions.svelte          # Game action buttons
│   │   └── Table.svelte            # Main game table
│   ├── stores/
│   │   ├── gameStore.ts            # Game state management
│   │   └── playerStore.ts          # Player state management
│   └── utils/
│       ├── cardUtils.ts            # Card calculation utilities
│       └── gameRules.ts            # Blackjack rules implementation
├── routes/
│   ├── +page.svelte                # Main game page
│   └── +layout.svelte              # App layout
├── app.html                        # HTML template
└── app.css                         # Global styles
```

## Getting Started

### Prerequisites

- Node.js (v16+)
- npm or yarn

## Installation

### 1. Clone the repository

```
git clone https://github.com/yourusername/blackjack-card-game.git
cd blackjack-card-game
```

### 2. Install dependencies

```
npm install
# or
yarn
```

### 3. Start the development server

```
npm run dev
# or
yarn dev
```

### 4. Open your browser to <http://localhost:5173>

## Development Guidelines

- Follow the established naming and coding conventions
- Create unit tests for game logic and utilities
- Ensure responsive design for both desktop and mobile
- Keep the UI clean and intuitive
- Handle API errors gracefully
- Implement proper state management

## Acknowledgments

- [Deck of Cards API](#) for providing the card deck functionality
- The SvelteKit team for the excellent framework