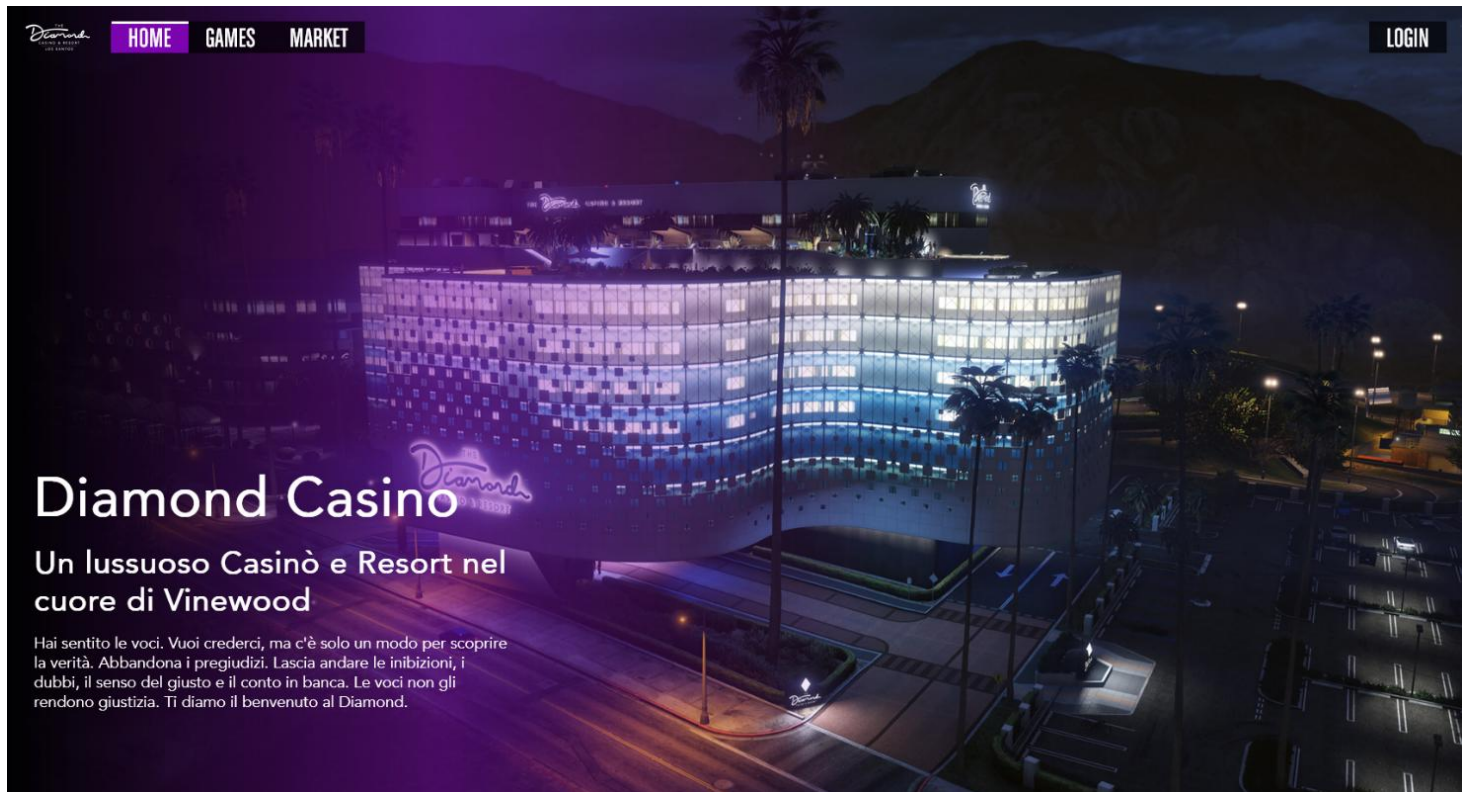


Relazione Tecnica

Progetto DIAMOND CASINO



1. Introduzione

Scopo del progetto

L'obiettivo era sviluppare una piattaforma ispirata al Diamond Casino di GTA Online, focalizzata su:

- Autenticazione utente con saldo persistente (*fish*).
- Giochi funzionali: Slot machine e Blackjack con logica di puntata.
- Gestione dinamica tramite sessioni PHP e chiamate AJAX.
- Database NoSQL (SQLite) per la persistenza dei dati.

Requisiti implementati

Requisito	Tecnologia	Dettaglio
Autenticazione	PHP Sessions	Cookie crittografati
Chiamate asincrone	Fetch API	Gestione saldo in tempo reale
Database	SQLite	Schema: username (UNIQUE), password (hash), fish (INT)
UI/UX	CSS Grid/Flexbox	Animazioni CSS3 per le slot

Tecnologie chiave:

- **Front-end:** HTML5 semantico, CSS3 (variabili custom, calc()), JavaScript (ES6+ con async/await).
- **Back-end:** PHP 8.0 (gestione sessioni, PDO per SQLite).
- **Strumenti:** Git per il versioning, Chrome DevTools per il debug delle chiamate API.

2. Idea e pianificazione

Genesi del progetto

L'idea è nata dall'analisi del sito ufficiale del Diamond Casino, con l'intento di replicarne:

- L'estetica con palette viola e nera.
- La fluidità delle animazioni .

Fasi di sviluppo

- **Setup iniziale (5 giorni):**
 - Configurazione ambiente XAMPP.
 - Creazione dello schema DB con setup_db.php.
- **Core functionality (12 giorni):**
 - Sistema di login/registrazione con validazione in real-time.
 - Implementazione giochi con saldo sincronizzato.
- **Refinement (4 giorni):**
 - Ottimizzazione mobile-first con CSS variables.
 - Debug delle sessioni PHP.

Divisione dei compiti

- **Francesco Callegaro:** Back-end, gestione account e index (PHP, sessioni, DB, HTML/CSS).
- **Sergio Piaser:** Front-end, giochi (HTML/CSS, animazioni JS).

3. Implementazione tecnica

Database SQLite

Struttura e operazioni CRUD:

```
1 // setup_db.php
2 $db->exec("CREATE TABLE users (
3     id INTEGER PRIMARY KEY AUTOINCREMENT,
4     username TEXT UNIQUE NOT NULL,
5     password TEXT NOT NULL,
6     fish INTEGER DEFAULT 0
7 );");
```

Sessioni PHP

Flusso di autenticazione:

- login_user.php: Verifica credenziali con password_verify().
- check_session.php: Middleware per pagine protette.

```
1 // login_user.php
2 if (password_verify($password, $user['password'])) {
3     $_SESSION['username'] = $user['username'];
4     $_SESSION['fish'] = $user['fish'];
5 }
```

Chiamate asincrone

Esempio: aggiornamento saldo nel Blackjack

```
1 // blackjack.js
2 async function updateFish() {
3     const response = await fetch('/php/set_fish.php', {
4         method: 'POST',
5         body: 'fish=${encodeURIComponent(newBalance)}'
6     });
7     if (!response.ok) throw new Error("API error");
8 }
```

Slot Machine: Logica di gioco

- Generazione simboli: Array con pesi per simboli rari.
- Calcolo vincite:

```
1 function checkWin(symbols) {
2     const diamonds = symbols.filter(s => s === '      ').length;
3     return diamonds > 0 ? diamonds * bet * 2 : 0;
4 }
```



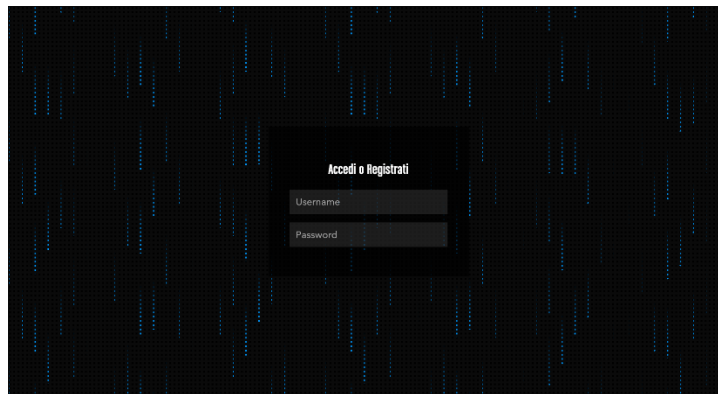
4. Difficoltà e soluzioni

Difficoltà 1: Sincronizzazione saldo

- **Sintomo:** Disallineamento tra saldo front-end e DB dopo multiple chiamate AJAX.
- **Debug:** Log delle transazioni con console.log e verifica dello stato della sessione.
- **Soluzione:** Usare le funzioni asincrone per leggere e settare il numero di fish dal database

Difficoltà 2: Gestione del database

- La gestione del database non è stata facile da implementare in quanto le competenze del team in questo ambito erano scarse.



```
1 // check_session.php
2 if (!isset($_SESSION['username'])) {
3     header("HTTP/1.1 403 Forbidden");
4     exit;
5 }
```

Testing

- **Unit test:** Verifica funzioni JS con console.assert().
- **Integration test:**
 - **Slot machine:** Autoclicker per 1.000+ giri con saldo iniziale infinito.

5. Conclusioni e miglioramenti

Valutazione finale

Successi:

- Sviluppo di un sito web completamente funzionante.
- Esperienza utente fluida grazie alle chiamate AJAX.

Limiti:

- Non c'è una scoreboard per vedere i punteggi degli altri giocatori.
- Mancanza di giochi multiplayer.
- Assenza dello storico delle transazioni.

Roadmap futura

- **Multiplayer:** Utilizzare WebSocket per il Poker in tempo reale.
- **Sicurezza:** Aggiungere 2FA con TOTP.
- **Nuovi giochi:** Aggiungere nuovi giochi per una varietà più ampia.
- **UI/UX:** Transizioni CSS per il cambio pagina.
- **Sviluppo UI/UX per mobile:** Sviluppo di un'interfaccia grafica comoda e ottimizzata per dispositivi mobile.