1. **Dokumentacja projektu:**

**Imię i nazwisko autora:** *Bartłomiej Florek*

**Numer albumu:** *125115*

**Temat projektu:** *gra w obstawianie drużyn*

**Nazwa przedmiotu:** *Technologie internetowe*

**Grupa laboratoryjna:** *lab1*

**Data oddania projektu:** *25.01.2024r.*

Strona tytułowa „MoonBet” przedstawia grę, w której można obstawiać mecze drużyn sportowych rywalizujących ze sobą. Strona obsługuje 6 różnych dyscyplin sportowych takich jak: piłkę nożną, boks, hokej, koszykówkę, rugby i krykiet. Gra polega na zaznaczaniu w wybranej dyscyplinie, drużyny, która według użytkownika wygra. Użytkownik może zadeklarować wartość jaką chce obstawić i na podstawie zaznaczonych drużyn może wygrać bądź przegrać.

*Spis treści*

1. *Dokumentacja projektu:*
2. *Opis założeń projektu:*
3. *Specyfikacja wymagań:*
4. *Testowanie*
5. *Opis techniczny projektu i dokumentacja kodu źródł.:*
6. *Opis kodu źródłowego oraz JSDOC:*
7. *Prezentacja warstwy użytkowej projektu*
8. *System kontroli wersji*
9. *Linki:*
10. **Opis założeń projektu:**

**Struktura strony:** strona składa się z głównego kontenera, w którym znajdują się poszczególne sekcje takie jak: nawigacja, header, footer, sekcja z grą oraz sekcja result, która wylicza kursy i wartość postawioną przez użytkownika.

**Saldo użytkownika:** użytkownik ma przypisane swoje saldo, które może wykorzystać do obstawienia. Użytkownik ma możliwość dodania środków, które program sprawdzi czy są poprawne tzn. czy nie są ujemne, albo czy użytkownik nie podał liter.

**Ligi, mecze i ich kursy:** ligi, mecze oraz kursy pobierane są z dostępnego w sieci API: <https://the-odds-api.com/liveapi/guides/v4/>. Każda drużyna ma swój odpowiadający kurs który determinuje jej szanse na wygraną.

**Interakcja z użytkownikiem:** użytkownik ma możliwość zaznaczenia drużyny, która według niego może wygrać, oraz obstawieniu pewnej wartości która po pomnożeniu przez łączy kurs, wylicza możliwą wygrana użytkownika.

**Zasady obstawiania:** użytkownik nie może postawić wartości, która jest ujemna, jest zerowa albo takiej, której nie dysponuje.

**Generowanie PDF:** użytkownik w podanym formularzu zaznacza opcje które go dotyczą, a następnie po kliknięciu w przycisk, program na podstawie zaznaczonych opcji generuje plik PDF na temat użytkownika i pobiera go.

**Estetyka:** zdefiniowano estetyczne style CSS, oraz efekty wizualne pokazujące zaznaczone mecze, sprawdzające błędną wartość, oraz podświetlanie pól po najechaniu kursorem.

**Responsywność i kompatybilność:** strona jest zoptymalizowana pod kątem różnych urządzeń, dzięki zastosowaniu responsywnego designu. Strona jest obsługiwana przez wiele przeglądarek.

1. **Specyfikacja wymagań:**

* **Wymagania funkcjonalne:**
* Aplikacja pobiera dane o meczach i ich kursach z zewnętrznego API
* Każdy mecz można obstawić, a kurs mu odpowiadający doda się do łącznego kursu
* Użytkownik ma możliwość zaznaczenia interesujących go meczy, które zwiększą jego kurs
* Po obstawieniu meczy użytkownik ma możliwość postawienia wartości
* Program wylicza możliwa wygraną na podstawie zaznaczonych meczy
* Program sprawdza czy użytkownik podał poprawne wartości zgodne z założeniami
* Aplikacja umożliwia dodanie środków do konta w przypadku gdyby zabrakło użytkownikowi
* Strona jest responsywna
* Aplikacja udostępnia formularz, w którym użytkownik podaje interesujące go dane
* Program na podstawie zaznaczonych danych generuje nowy plik PDF z informacjami o użytkowniku
* Interfejs jest responsywny przez co dostosowuje się do mniejszych ekranów
* Interfejs posiada atrakcyjny wizualnie wygląd z zastosowaniem stylów CSS.
* **Wymagania niefunkcjonalne:**
* Strona jest kompatybilna z najpopularniejszymi przeglądarkami: Chrome, Firefox, Safari, Brave itp.
* Strona ładuje się szybko aby zapewnić natychmiastowy dostęp do gry.
* Projekt jest zgodny z obowiązującymi standardami internetowymi i spełnia wymagania dotyczące semantyki HTML5, CSS3 oraz zaleceń dostępności
* Mecze w grze zgadzają się z aktualnymi meczami na obecny dzień
* Aplikacja tworzona w środowisku VSCode

1. **Testowanie**

Wprowadzanie danych przez użytkownika:

Wprowadzanie liter do dodawania środków:

Obraz zawierający tekst, zrzut ekranu, linia, Czcionka

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznieObraz zawierający tekst, Czcionka, design, przewodnik

Opis wygenerowany automatycznie

Wprowadzanie wartości ujemnych:

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznieObraz zawierający tekst, Czcionka, design, przewodnik

Opis wygenerowany automatycznie

Wprowadzenie wartości poprawnych:

Obraz zawierający Czcionka, tekst, zrzut ekranu, Grafika

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznieObraz zawierający tekst, Czcionka, zrzut ekranu, Grafika

Opis wygenerowany automatycznie

Jeśli nie obstawimy żadnego meczu i nasz kurs będzie wynosić 0, jeśli postawimy ujemne wartości lub jeśli postawimy 0zł, program zwróci błąd:

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

W przypadku próby obstawienia kwoty której nie mamy program również zwróci błąd:

Obraz zawierający Czcionka, tekst, zrzut ekranu, Grafika

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, Czcionka, logo

Opis wygenerowany automatycznie

1. **Opis techniczny projektu i dokumentacja kodu źródł.:**

* Środowisko pracy nad projektem JavaScript: VSCode: 1.85.2 (user setup)

1. **Opis kodu źródłowego oraz JSDOC:**

* Asynchroniczne ładowanie body, które dodaje do divów datę dzisiejszą oraz jutrzejszą
* Funkcja add\_funds(), która odpowiada za dodawanie środków do salda użytkownika. Sprawdza czy użytkownik podał odpowiednie dane, które mogą zasilić saldo.
* Funkcja toggleDiv(element), odpowiada za zaznaczanie divow oraz odznaczanie ich w celu pobrania wartości kursu
* addEventListener(„input”, function(){}, odpowiada za obliczanie stawki użytkownika \* łączy kurs wszystkich zaznaczonych meczów. Nasłuchuje kiedy przycisk input zostaje wciśnięty i wtedy oblicza.
* Funkcja submit(), jej celem jest pobranie łącznego kursu użytkownika oraz stawki i sprawdzenie czy gracz spełnia wymogi aby obstawić grę.
* Funkcja question() odpowiada za otworzenie dialogowego okna formularza
* Funkcja closeDialog() zamyka okno dialogowe formularza
* Funkcja generatePdf(event), pobiera dane z formularza, dodaje je do nowo utworzonej strony, która dzięki wtyczce html2pdf() zamienia ja na plik PDF i pobiera go
* Function myhref(web) odpowiada za nawigacje do odpowiednio podanej strony

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, dokument

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie Obraz zawierający tekst, zrzut ekranu, dokument, Czcionka

Opis wygenerowany automatycznie Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

1. **Prezentacja warstwy użytkowej projektu**

Na obrazu poniżej przedstawiono główne okno aplikacji. Użytkownikowi po uruchomieniu strony pokazują się takie elementy jak: liga, w której odbywają się poszczególne mecze, drużyny, które ze sobą rywalizują jak i ich przewidywana szansa na wygraną.

Obraz zawierający zrzut ekranu, tekst, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Użytkownik może zaznaczać, która drużyna według niego wygra, a odpowiadający jej kurs doda się do odpowiedniego pola z aktualnym kursem.

Obraz zawierający zrzut ekranu, tekst, linia, Prostokąt

Opis wygenerowany automatycznie

W końcowej fazie użytkownik może zadeklarować kwotę jaką chce założyć, a program automatycznie wyliczy mu możliwą wygraną i sprawdzi czy użytkownik dysponuje odpowiednią ilością środków.

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, Czcionka, logo

Opis wygenerowany automatycznie

Obraz zawierający tekst, Czcionka, zrzut ekranu, Grafika

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, Czcionka, Grafika

Opis wygenerowany automatycznie

Jeśli użytkownikowi skończą się dostępne środki, możliwe jest dodane nowej ilości, poprzez kliknięcie w napis „dodaj środki”:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Po kliknięciu użytkownik zostaje poproszony o podanie kwoty jaką chce dodać:

Obraz zawierający tekst, zrzut ekranu, linia, Czcionka

Opis wygenerowany automatycznie

Program sprawdza czy podana kwota jest odpowiednia, i w przypadku błędnego wprowadzenia danych, odrzuci możliwość dodania środków

Wprowadzanie wartości ujemnych:

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznieObraz zawierający tekst, Czcionka, design, przewodnik

Opis wygenerowany automatycznie

Wprowadzanie liter:

Obraz zawierający tekst, zrzut ekranu, linia, Czcionka

Opis wygenerowany automatycznieObraz zawierający tekst, Czcionka, design, przewodnik

Opis wygenerowany automatycznie

Wprowadzenie poprawnych danych:

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznieObraz zawierający tekst, Czcionka, zrzut ekranu, Grafika

Opis wygenerowany automatycznie

1. **System kontroli wersji**

Projekt realizowany był z wykorzystaniem systemu kontroli wersji Github, wszystkie pliki źródłowe projektu znajdują się pod adresem: <https://github.com/flor3kk/ProjektTI>. Link do strony: <https://flor3kk.github.io/ProjektTI/pilka.html> Poniżej przedstawiono zawartość folderu.

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

1. **Linki:**

* <https://the-odds-api.com/>
* <https://www.geeksforgeeks.org/how-to-use-the-javascript-fetch-api-to-get-data/>
* <https://stackoverflow.com/questions/72596411/how-to-fetch-api-and-display-data-in-my-html-code>
* <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/dialog>
* <https://www.w3schools.com/howto/howto_js_toggle_hide_show.asp>
* <https://www.geeksforgeeks.org/how-to-design-runtime-generated-pdf-via-html/>
* <https://www.w3schools.com/jsref/met_win_open.asp>