Brain Blitz

"Il gioco è la chiave dell'apprendimento."

Indice

- L'idea iniziale...
- Features per gli utenti
- Modalità di gioco
- Mock-up
- Tecnologie utilizzate
- Pacchetti installati
- Struttura e installazione del progetto
- API endpoints
- Database
- Strategie di gaming, accessibilità, teoria dei colori e visualizzazione dati
- Sicurezza
- Documentazione
- Prossimi passi...

L'idea iniziale...

La primissima idea era quella di creare un **gioco a quiz** su un server web con Node, Vite, Vue ed Express. Utilizzando un **database di supporto** avremmo gestito l'accesso degli utenti, i rapporti di amicizia tra di loro e tutte le domande con relative risposte. Non ci siamo discostati troppo dalla nostra idea di partenza, ma abbiamo aggiunto mano a mano delle features per renderlo più piacevole e accattivante.
A progetto terminato, gli utenti accedono tramite un **username univoco** e una password; il profilo può successivamente essere personalizzato con un'immagine. È possibile inviare **richieste di amicizia** ad altri giocatori conoscendone l'username. Ci saranno **più modalità di gioco** per rendere sempre nuovo e interessante il sito. Le domande sono divise in categorie in base all'argomento e sono di due tipi: scelta multipla e vero/falso. Ogni domanda possiede una spiegazione aggiuntiva alla risposta corretta.
Un partita è composta da 5 domande e i punteggi sono aggiornati in **tempo reale** alla fine della stessa.

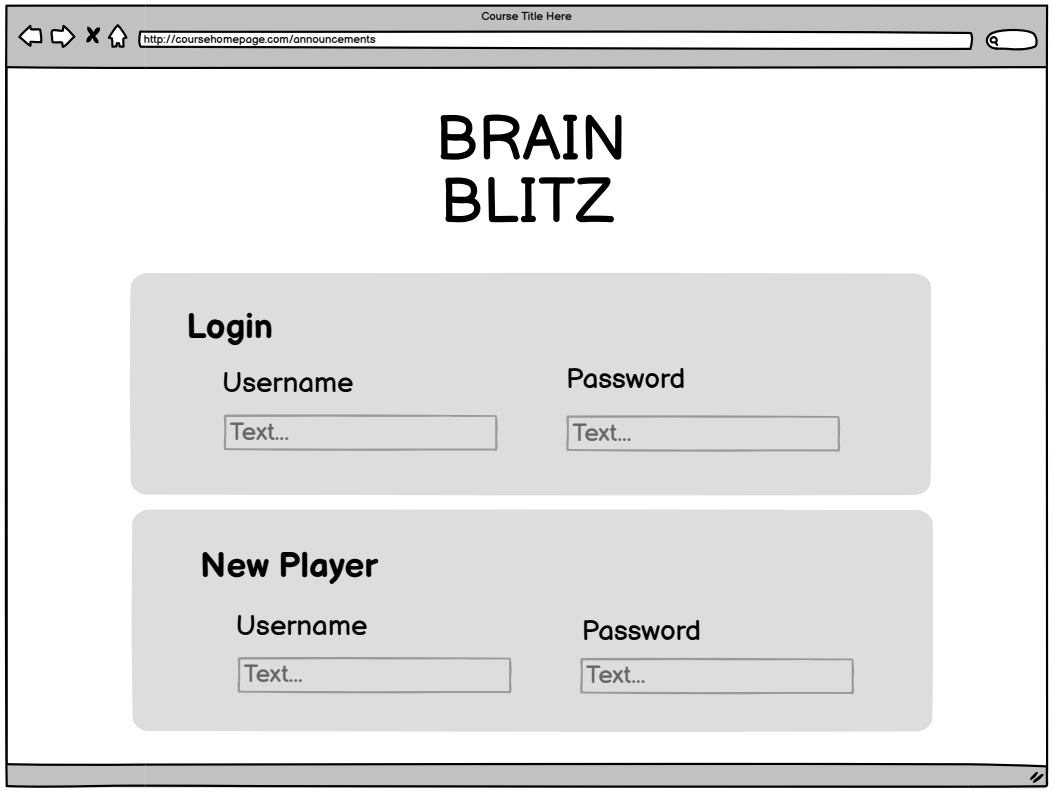
Features per gli utenti

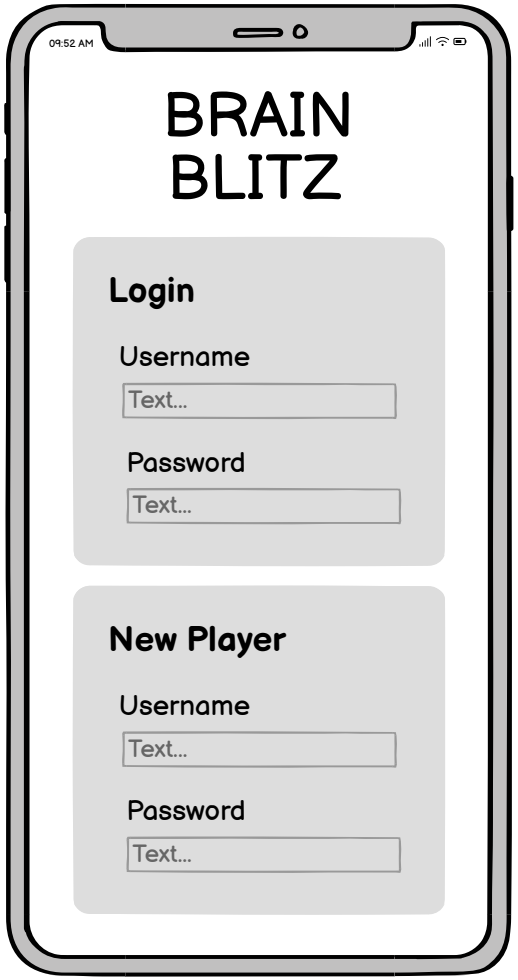
- Possibilità di registrarsi, fare l'accesso e **personalizzare** il proprio profilo tramite immagine e username;
- Visualizzazione delle proprie **statistiche personali**, come vittorie e partite giocate;
- Possibilità di creare una propria **lista di amici**, tramite apposite richieste di amicizia;
- **Badges** da sbloccare completando svariate missioni; ad ogni badge corrisponde un gadget che può arricchire ulteriormente il profilo;
- Più modalità di gioco: **allenamento**, **gioco online** e **sfida**;
- Tre stati dell'utente: **online** (l'utente è nella home e può ricevere richieste di sfida), **offline** (l'utente è uscito dal sito) ed **occupato** (l'utente è in partita e non può ricevere richieste di sfida).

Modalità di gioco

Il gioco si divide in:

- **OFFLINE (CON spiegazioni e senza limiti di tempo):**
  - **allenamento** = una modalità che permette al giocatore di prendere confidenza con le diverse categorie, che possono essere scelte all'inizio di ogni partita, e di visualizzare le relative spiegazioni.
- **ONLINE (SENZA spiegazioni e con un limite di tempo prestabilito):**
  - **gioco online** = una richiesta di gioco verso altri utenti casuali online che permette di mettersi alla prova dopo la modalità di allenamento;
  - **sfida tra amici** = una richiesta di gioco verso un utente specifico appartenente alla lista di amici.



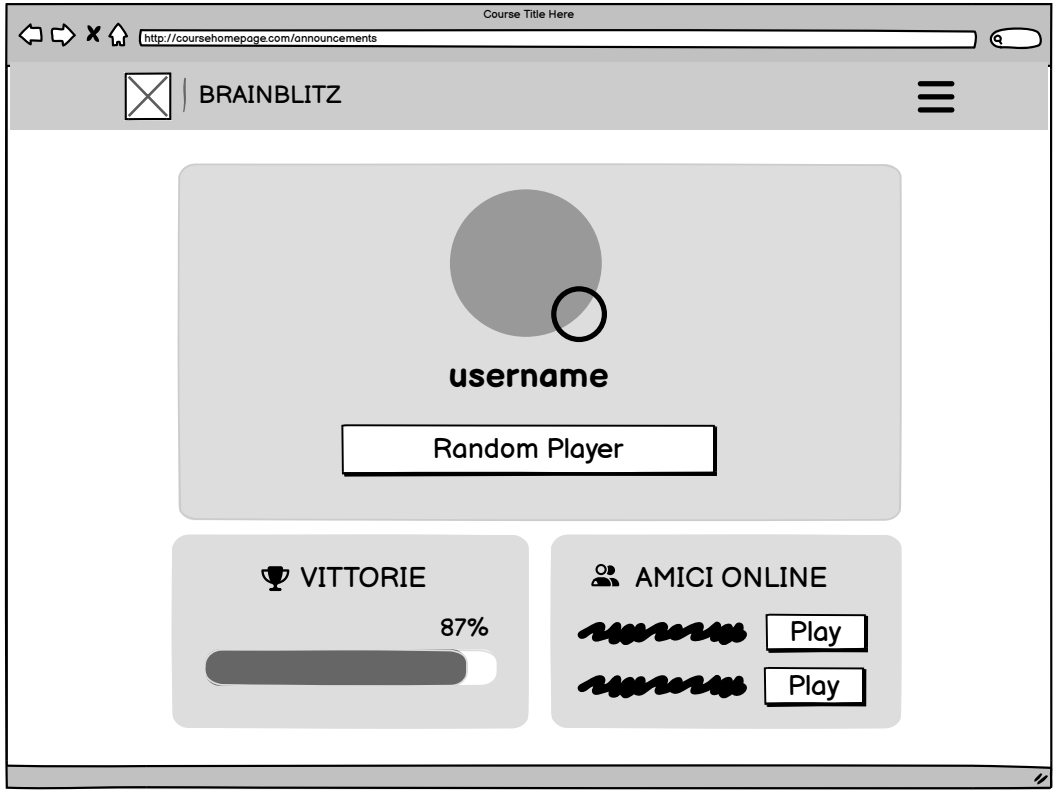


Il login rimane fisso dalla prima volta che viene eseguita la registrazione a meno che non si compia Logout

IMPOSAZIONI

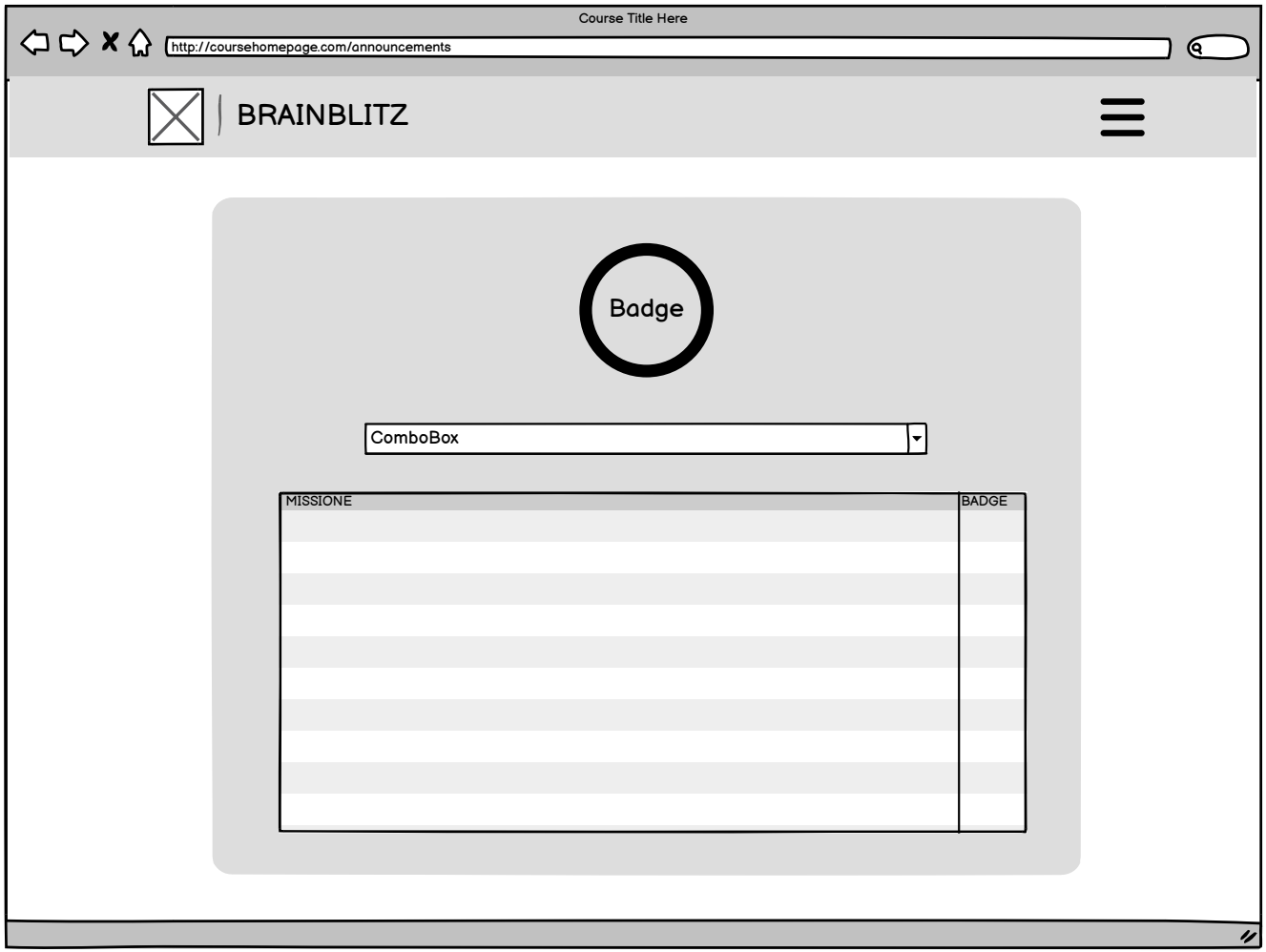
AMICI

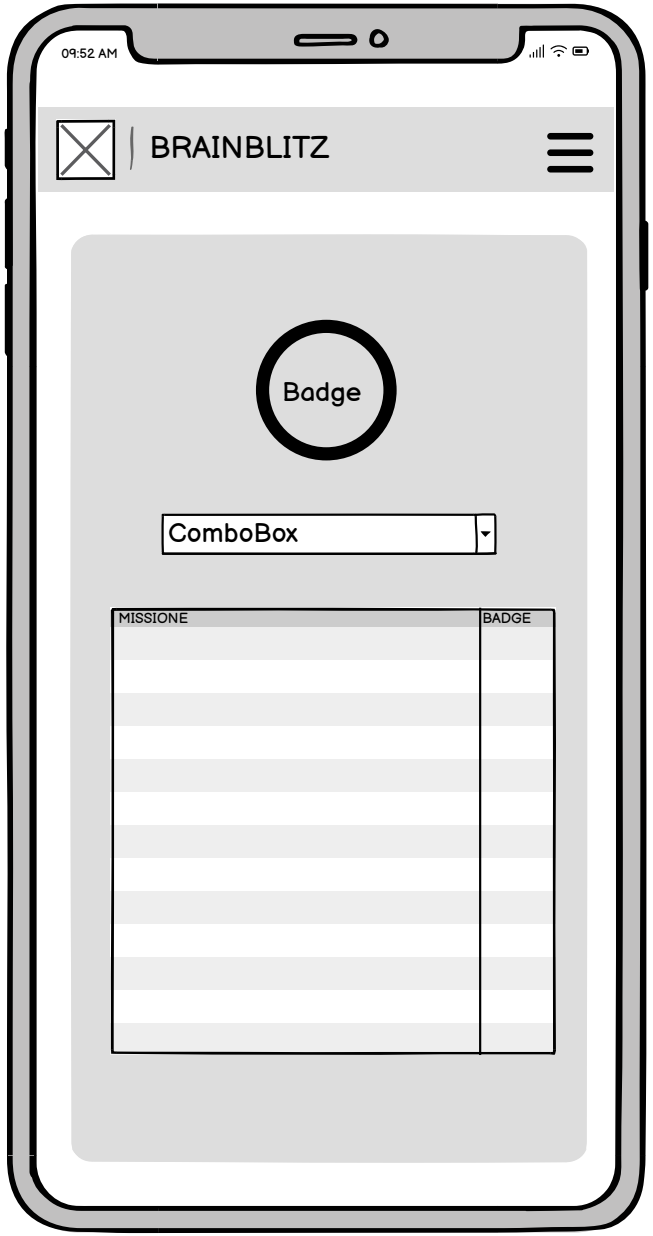
HOME

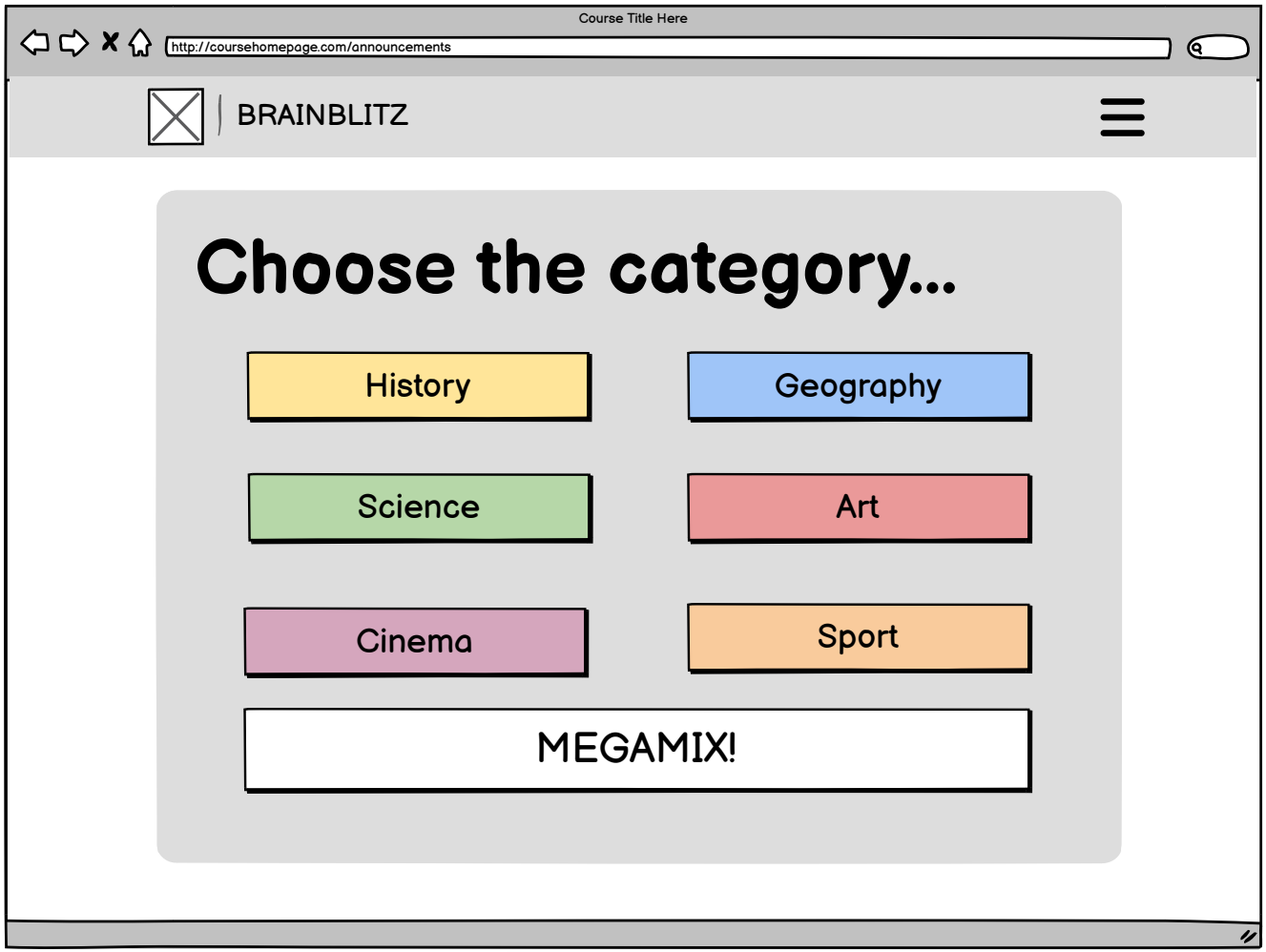


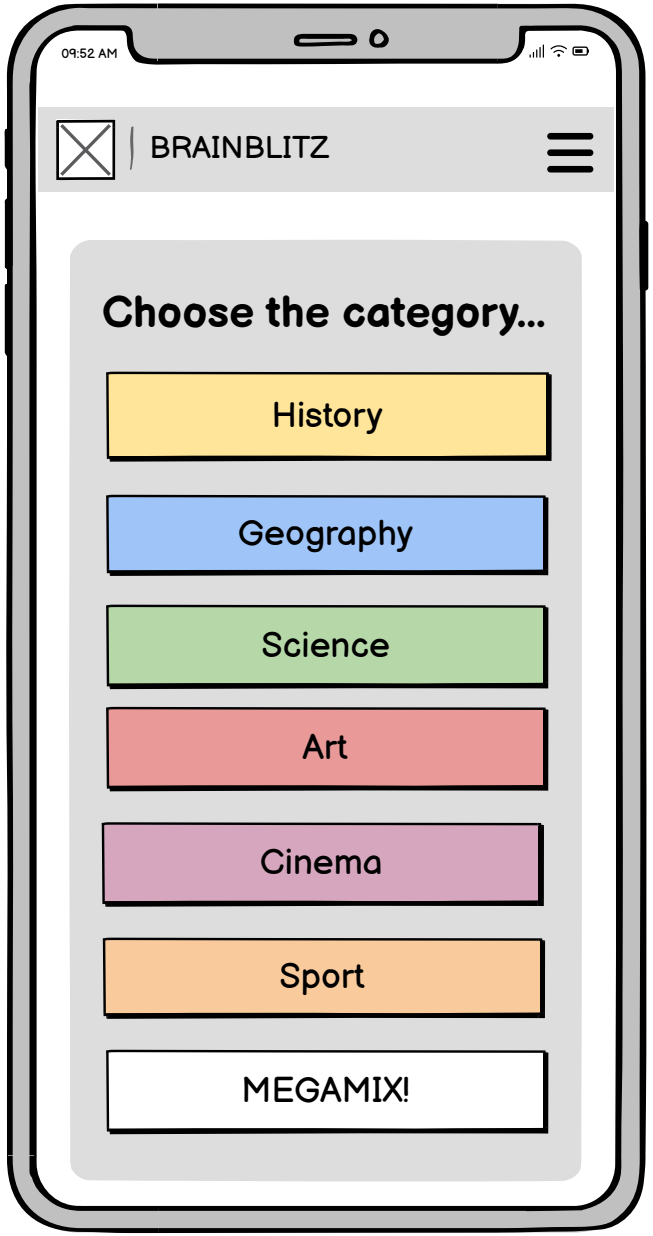
AMICI ONLINE

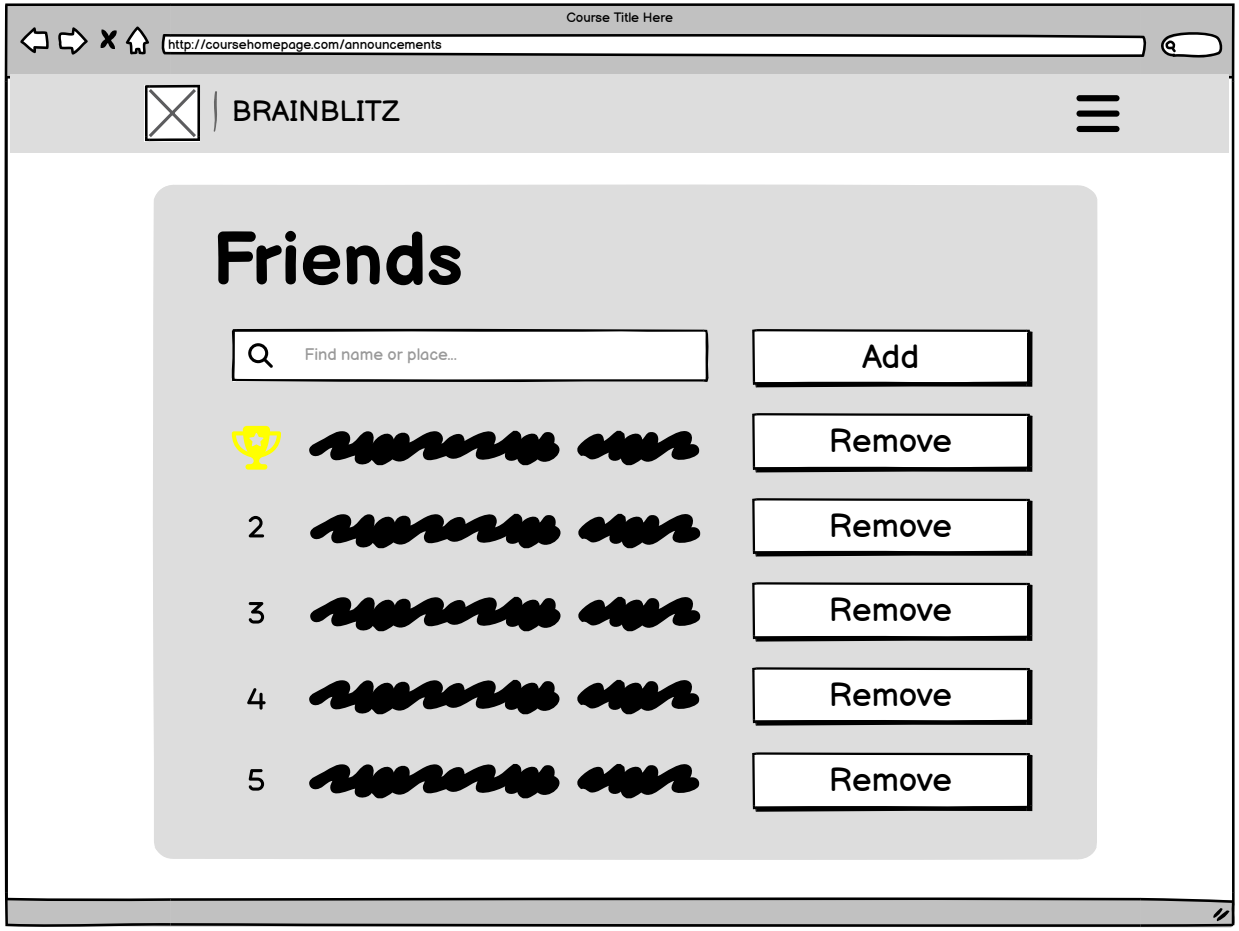
no one is here...


Course Title Here
http://coursehomepage.com/announcements
BRAINBLITZ
Badge
ComboBox
MISSIONE
BADGE


09:52 AM
BRAINBLITZ
Badge
ComboBox
MISSIONE
BADGE

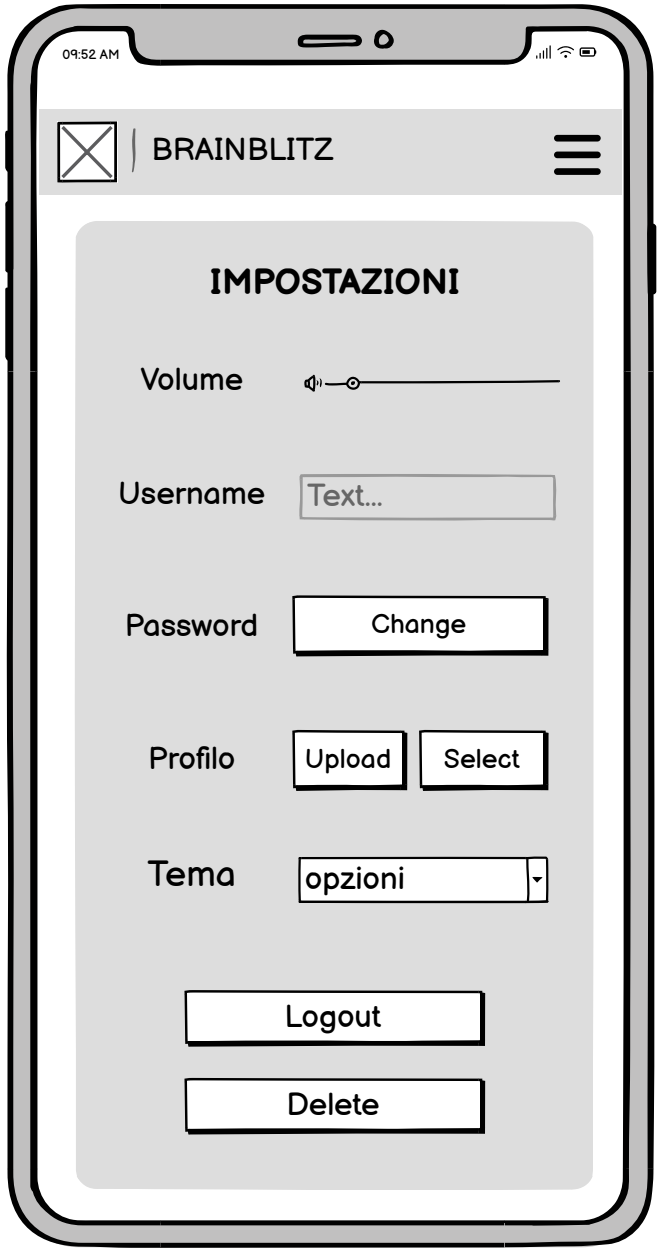

Course Title Here
http://coursehomepage.com/announcements
BRAINBLITZ
Choose the category...
History
Geography
Science
Art
Cinema
Sport
MEGAMIX!

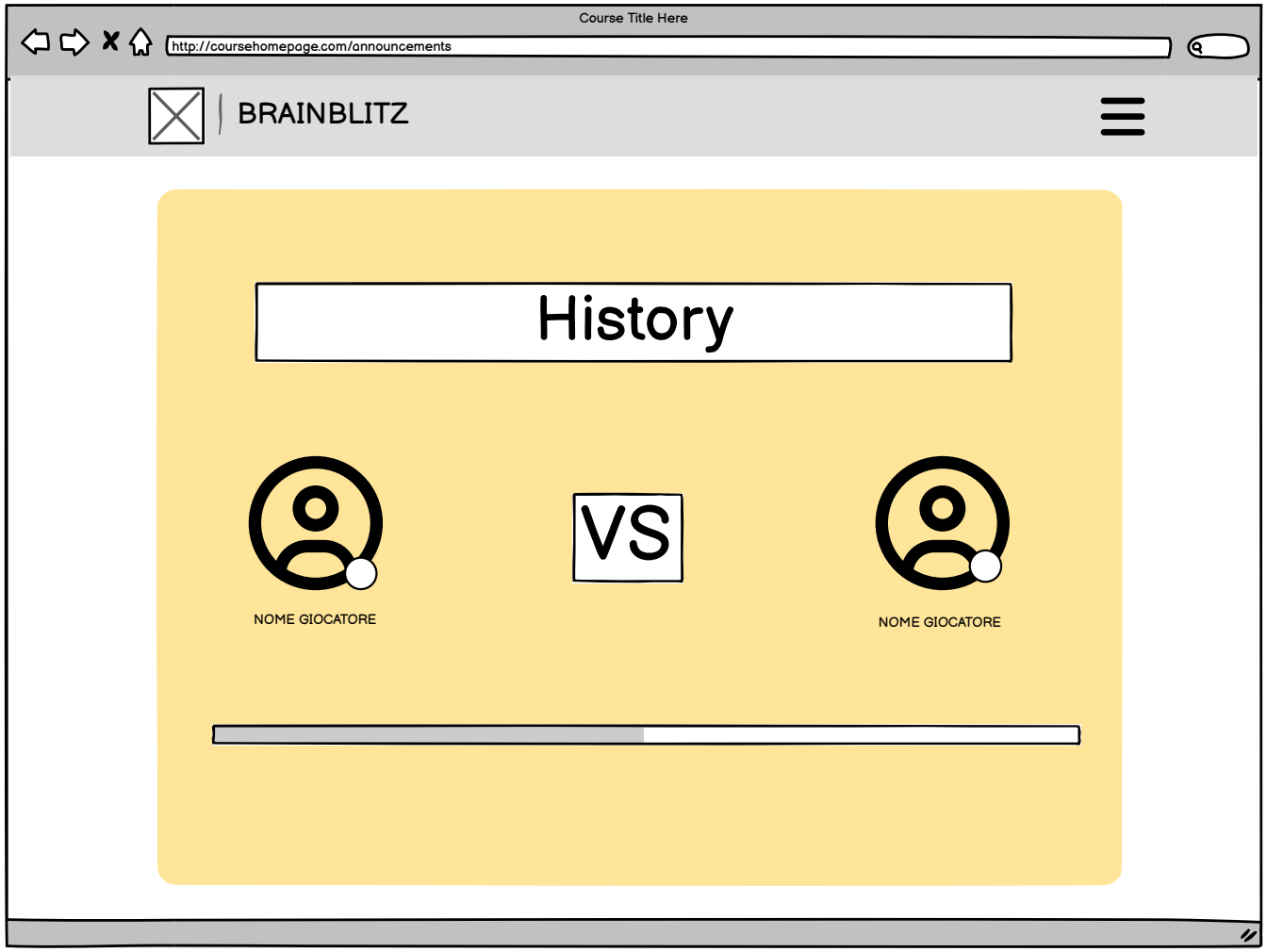

09:52 AM
BRAINBLITZ
Choose the category...
History
Geography
Science
Art
Cinema
Sport
MEGAMIX!

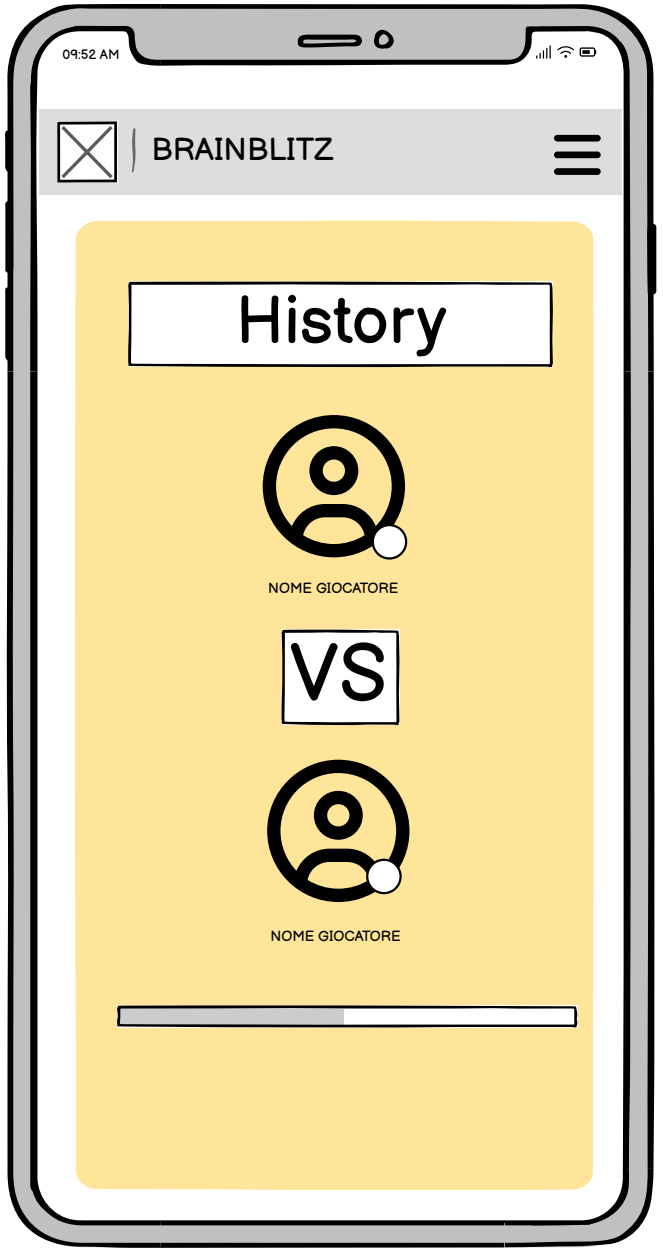

Course Title Here
http://coursehomepage.com/announcements
BRAINBLITZ
Friends
Find name or place...
Add
Remove
2
Remove
3
Remove
4
Remove
5
Remove

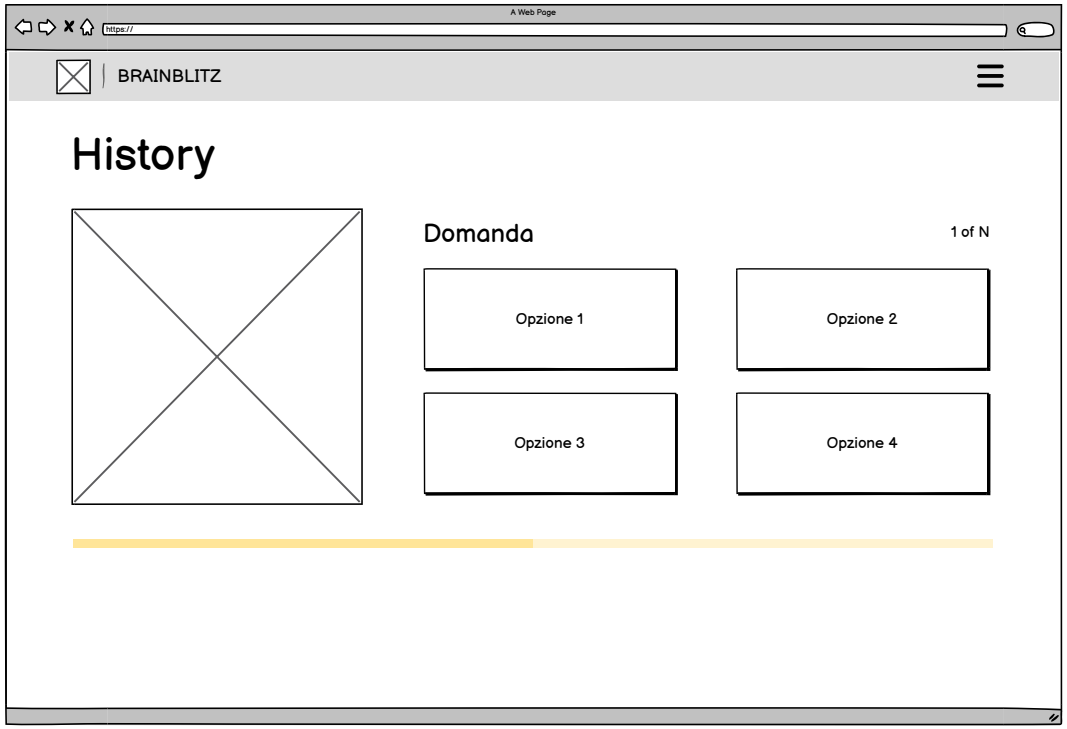

09:52 AM
BRAINBLITZ
Friends
Find name or place...
Add
Remove
2
Remove
3
Remove
4
Remove
5
Remove
5
Remove

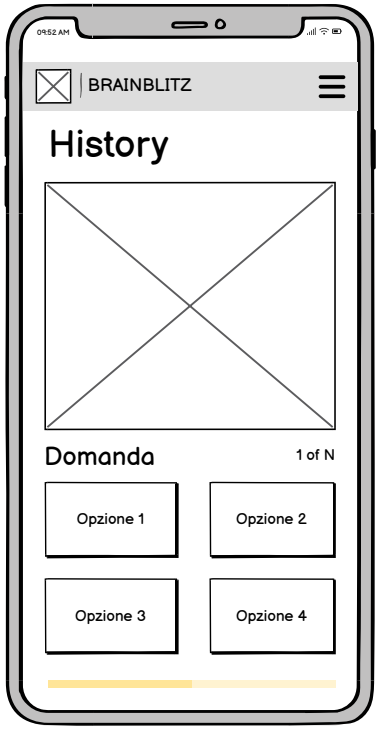

Course Title Here
http://coursehomepage.com/announcements
BRAINBLITZ
IMPOSTAZIONI
Volume
Username
Text...
Password
Change
Profilo
Upload
Select
Tema
opzioni
Logout
Delete

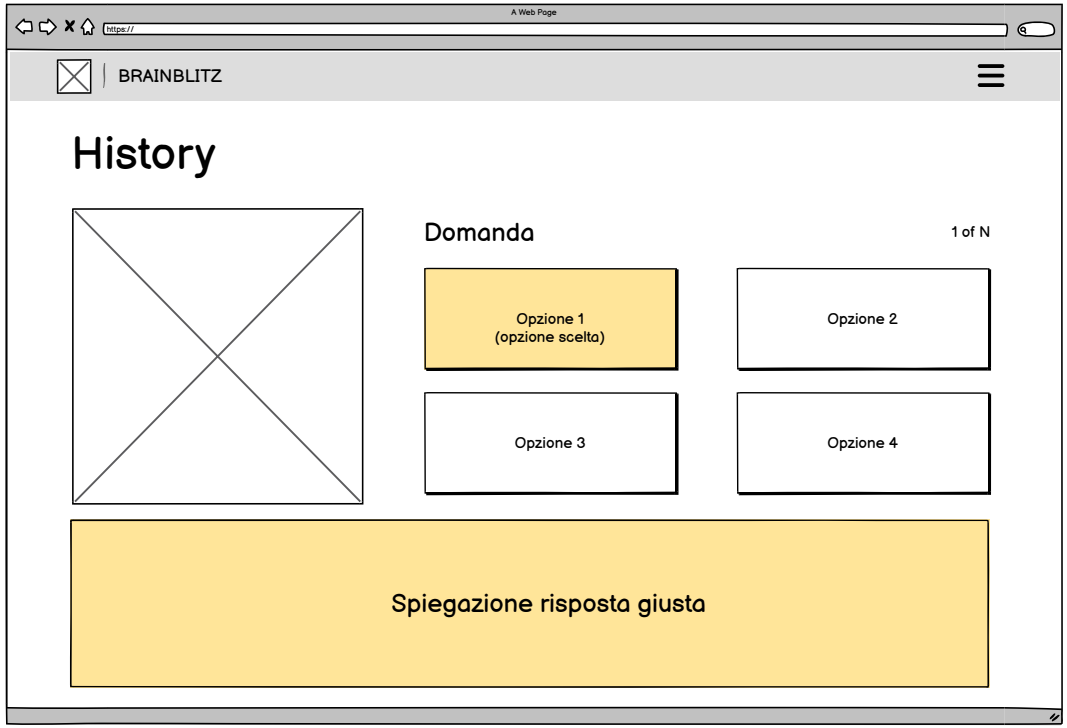

09:52 AM
BRAINBLITZ
IMPOSTAZIONI
Volume
Username
Text...
Password
Change
Profilo
Upload
Select
Tema
opzioni
Logout
Delete


Course Title Here
http://coursehomepage.com/announcements
BRAINBLITZ
History
VS
NOME GIOCATORE
NOME GIOCATORE

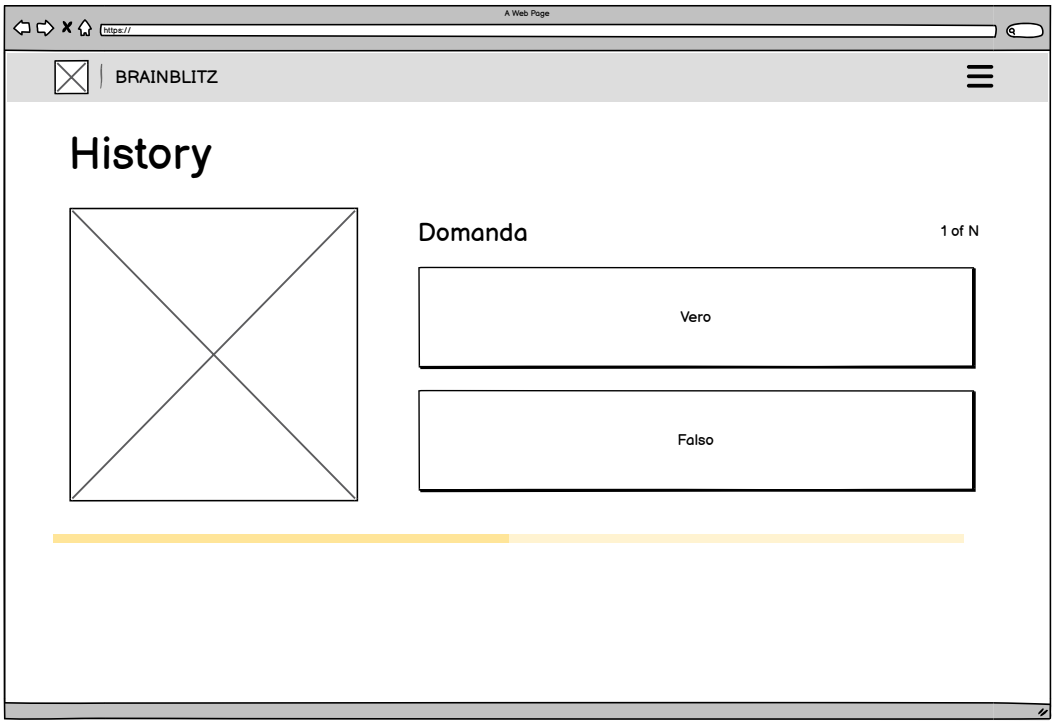

09:52 AM
BRAINBLITZ
History
NOME GIOCATORE
VS
NOME GIOCATORE

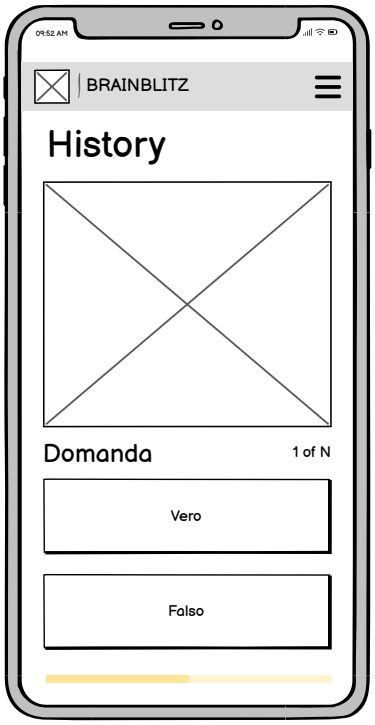

A Web Page
https://
BRAINBLITZ
History
Domanda
1 of N
Opzione 1
Opzione 2
Opzione 3
Opzione 4

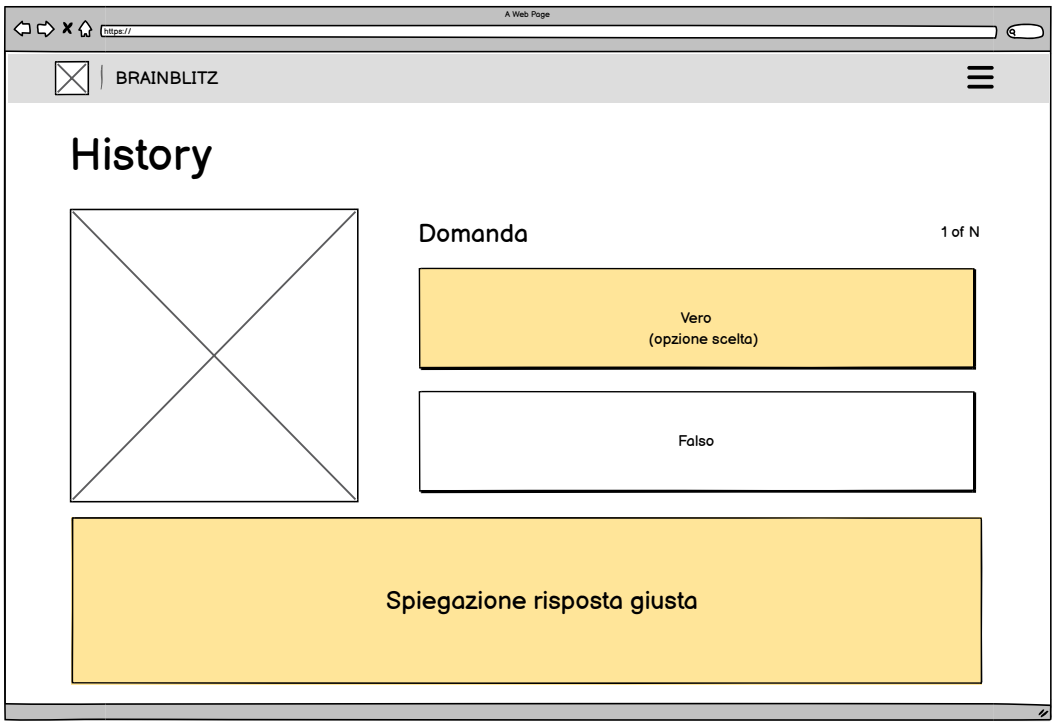

09:52 AM
BRAINBLITZ
History
Domanda
1 of N
Opzione 1
Opzione 2
Opzione 3
Opzione 4


A Web Page
https://
BRAINBLITZ
History
Domanda
1 of N
Opzione 1
(opzione scelta)
Opzione 2
Opzione 3
Opzione 4
Spiegazione risposta giusta

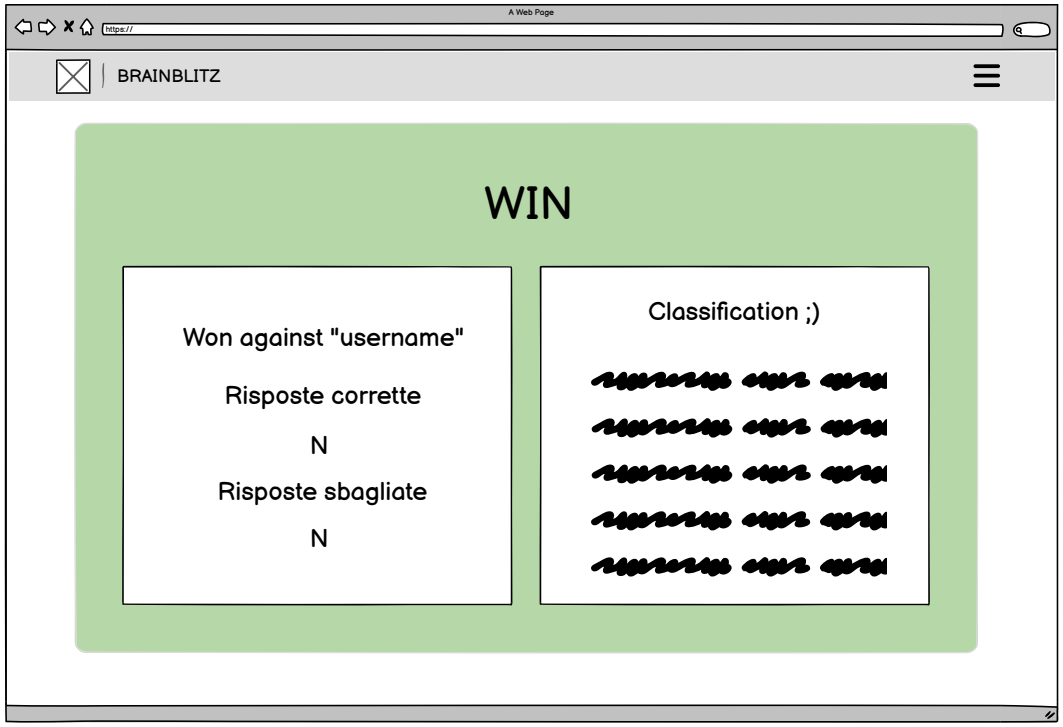

09:52 AM
BRAINBLITZ
History
Spiegazione risposta giusta
Domanda
1 of N
Opzione 1
(opzione scelta)
Opzione 2
Opzione 3
Opzione 4

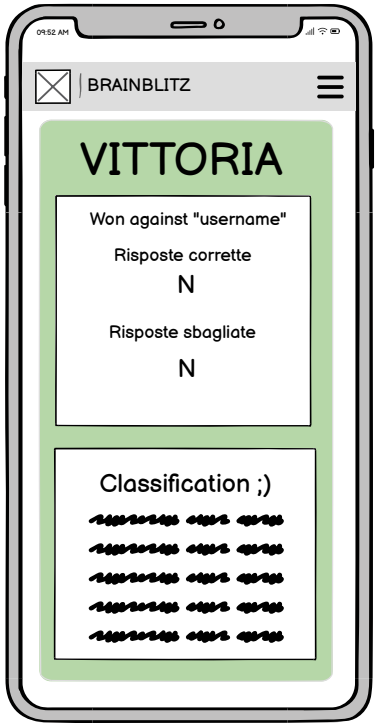

A Web Page
https://
BRAINBLITZ
History
Domanda
1 of N
Vero
Falso

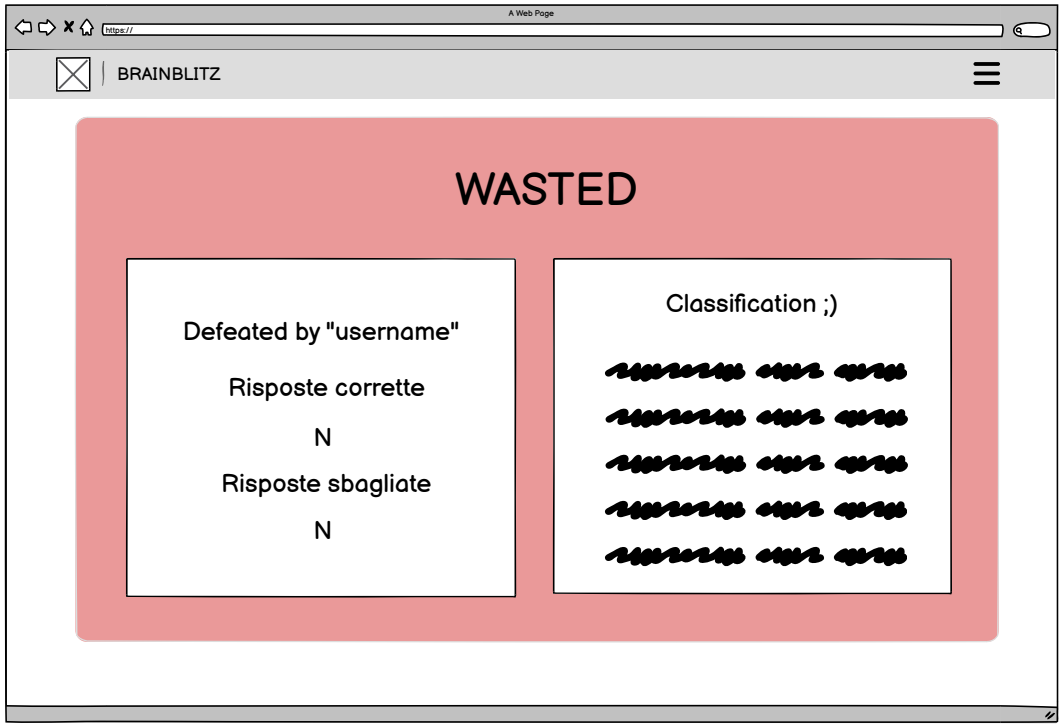

09:52 AM
BRAINBLITZ
History
Domanda
1 of N
Vero
Falso

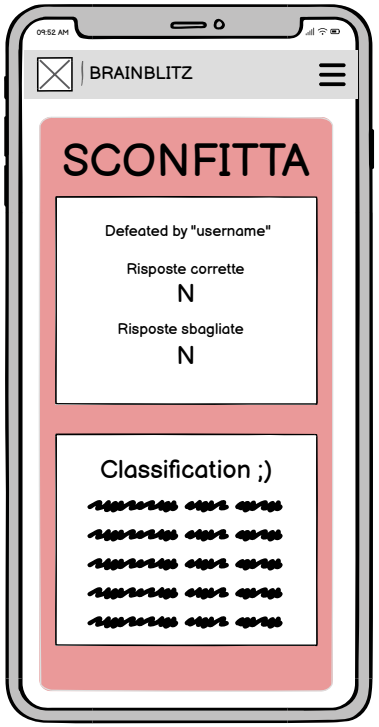

A Web Page
https://
BRAINBLITZ
History
Domanda
1 of N
Vero
(opzione scelta)
Falso
Spiegazione risposta giusta


09:52 AM
BRAINBLITZ
History
Spiegazione risposta giusta
Domanda
1 of N
Vero (opzione scelta)
Falso


A Web Page
https://
BRAINBLITZ
WIN
Won against "username"
Risposte corrette
N
Risposte sbagliate
N
Classification ;)


09:52 AM
BRAINBLITZ
VITTORIA
Won against "username"
Risposte corrette
N
Risposte sbagliate
N
Classification ;)


A Web Page
https://
BRAINBLITZ
WASTED
Defeated by "username"
Risposte corrette
N
Risposte sbagliate
N
Classification ;)


09:52 AM
BRAINBLITZ
SCONFITTA
Defeated by "username"
Risposte corrette
N
Risposte sbagliate
N
Classification ;)

Tecnologie utilizzate

- **XAMPP** (per avviare MySQL e accedere al database) = ci ha permesso di testare il sito localmente e di creare il nostro database con i relativi dati;
- **Node.js** = ambiente di sviluppo per creare web server;
- **npm** (Node Package Manager) = gestore di librerie e funzioni di Node;
- **Vite** = un build tool per incapsulare Vue all'interno di HTML;
- **Vue.js** = framework JS per costruire un'istanza dell'app renderizzata con mount;
- **Vue router** = libreria di routing per Vue.js per gestire la navigazione tra pagine;
- **Axios** = una libreria per effettuare richieste HTTP e per interagire con API e server;
- **TypeScript** = un super set (estensione di JS) che aggiunge i tipi statici (number, string, ...);
- **Express** = un framework server-side per web app su Node;
- **Bootstrap** e **CSS** = linguaggi di stile (Bootstrap è un framework CSS) per descrivere l'aspetto di pagine web;
- **Socket** = un endpoint per consentire la comunicazione client-server in tempo reale;
- **Visual Studio Code (Live share)** = ci ha permesso di lavorare insieme anche quando non eravamo fisicamente nello stesso luogo, in particolare Live Share ha fatto in modo di mostrare file condivisi e cambiamenti apportati in tempo reale.

Pacchetti installati

- **connection-history-api-fallback** = un middleware utilizzato per le SPA e la gestione la navigazione lato client;
- **body-parser** = un middleware Node.js per convertire dati della richiesta in un oggetto JS;
- **vue-clerk** (Software Development Kit) = un SDK per integrare Clerk;
- **DOMPurify** = una libreria per prevenire attacchi hacker, rimuovendo il contenuto pericoloso negli input (noi l'abbiamo utilizzato nella barra di ricerca degli amici);
- **socket.io** = libreria sopra WebSocket con funzionalità aggiuntive;
- **mysql2** = client SQL per Node.js;
- **http** = protocollo per la gestione di trasmissione di documenti ipertestuali.

Struttura e installazione del progetto

Il progetto si suddivide in due cartelle:

- **BackEnd** = realizzato in ambiente Node tramite Express, MySQL e Typescript;
- **FrontEnd** = realizzato tramite Vite, Vue, Axios, Typescript.

Come installare il progetto:

1. Clonare la cartella di GitHub: ' https://github.com/GbriRci/BrainBlitz '
2. Installare le dipendenze tramite il comando ' *npm i* '
3. Inserire all'interno di XAMPP il database contenuto nella cartella *backend/utils/database.txt* e avviare Apache e MySQL
4. Aprire due terminali nelle cartelle Frontend e BackEnd e lanciare il comando ' *npm run dev* '
5. Visitare il link: ' http://localhost:5173/accesso '
6. Fare l'accesso con username: ' chiara ' e password: ' 5febbraio25 '

API Endpoint

1. **AMICI**
   - **GET**
     - */api/amici/:username*: prende tutti gli amici di un determinato utente;
     - */api/amici/ricerca/:username*: ricerca un utente dato un username;
     - */api/amici/richieste/:username*: prende le richieste di amicizia dato un utente;
     - */api/amici/giarichiesto/:username*: evita che si compiano più richieste di amicizia allo stesso utente;
     - */api/amici/online/:user*: restituisce solo gli amici online;
   - **DELETE**
     - */api/amici/cancella/:user/:amico*: eliminare un utente dalla lista degli amici;
     - */api/amici/rifiuta/:user/:richiesta*: rifiutare una richiesta di amicizia;
   - **POST**
     - */api/amici/richiesta*: inviare una richiesta di amicizia;
     - */api/amici/accetta*: accettare una richiesta di amicizia;

2. **BADGE**
   - **GET**
     - */api/badge*: restituisce tutti i badge;
     - */api/badge/sbloccati/:Username*: restituisce tutti i badge sbloccati;
   - **PATCH**
     - */api/badge/equipaggiato*: aggiorna il badge equipaggiato di un utente;
3. **CATEGORIE**
   - **GET**
     - */api/categoria/:IDCategoria*: restituisce i dati di una categoria dato l'id;
     - */api/categorie*: restituisce tutte le categorie;
4. **DOMANDA**
   - **GET**
     - */api/domanda/:IDCategoria*: prende una domanda casuale da una categoria specifica;
     - */api/domanda*: prende una domanda casuale tra tutte le categorie;
5. **REGISTRAZIONE**
   - **POST**
     - */api/registrazione/utente*: inserisce un nuovo utente nel database;
6. **RISPOSTE**
   - **GET**
     - */api/risposte/:IDDomanda*: prende le risposte di una data domanda;
7. **UTENTE**
   - **GET**
     - */api/utente/:username*: restituisce i dati di un utente;
     - */api/utente/sfide/:username*: restituisce le sfide ricevute da un utente;
     - */api/utente/badge/:username*: restituisce i badge sbloccati da un utente;
   - **PATCH**
     - */api/utente/statistiche*: aggiorna il numero di partite giocate di un utente;
     - */api/utente/statistiche/vittoria*: aggiorna il numero di vittorie di un utente;
     - */api/utente/online*: rende un utente ' online ';
     - */api/utente/disconnessione*: rende un utente ' non online ';
     - */api/utente/occupato*: rende un utente ' occupato ';
     - */api/utente/nonoccupato*: rende un utente ' non occupato ';
   - **POST**
     - */api/utente/richiesta*: invia una richiesta di gioco ad un utente;
     - */api/utente/achievement/:user/:IDBadge*: aggiunge un nuovo badge sbloccato ad un utente;
   - **DELETE**
     - */api/utente/rifiuta/:user/:sfida*: rifiuta un richiesta di gioco.

Database

Come database abbiamo scelto **mySQL** in base alle nostre conoscenze pregresse; la struttura è stata completamente progettata e realizzata da noi (incluse le query).
Abbiamo creato le seguenti tabelle:

- **ACHIEVEMENT** (per memorizzare i badge sbloccati dall'utente);
- **AMICI** (lista di amicizie presenti nel server);
- **BADGE** (lista di badge con le relative missioni per ottenerli);
- **CATEGORIE** (lista delle categorie delle domande);
- **DOMANDE** (liste delle domande con relativa spiegazione);
- **RICHIESTE** (lista delle richieste di amicizia non ancora accettate/rifiutate);
- **RISPOSTE** (lista delle risposte per ogni domanda, divise tra corrette e sbagliate);
- **SFIDE** (lista delle sfide inviate agli amici da ogni utente non ancora accettate/rifiutate);
- **UTENTE** (dati dell'utente e statistiche personali).

Strategie di gaming, accessibilità, teoria dei colori e visualizzazione dati

- **STRATEGIE DI GAMING:**
  Il target del nostro sito potrebbe spaziare tra un pubblico annoiato alla ricerca di **stimoli divertenti** a uno interessato ai **vantaggi educativi** che si possono fruire dallo studio unito a elementi ludici; sono infatti presenti svariate strategie di gaming che catturano l'attenzione dell'utente portandolo a giocare, e '*rigiocare*' ancora, che possono essere molto utili nel campo dell'educazione:
    - possibilità di sbloccare **badge** al compimento di missioni;
    - **contatori** per le partite giocate e per il numero di vittorie;
    - possibilità di **confrontarsi** con i propri amici;
    - più **modalità di gioco** che aumentano la variabilità.

  L' idea è quella di creare un database con domande e risposte compilate in base alle esigenze didattiche e distribuire il sito al suo target.
- **ACCESSIBILITÀ:**
  Per rendere il sito di più facile comprensione a persone con condizioni visive che alterano la percezione del colore, abbiamo inserito nelle varie risposte delle domande **icone** differenti per simboleggiare le risposte corrette/sbagliate. Precedentemente il messaggio era prima veicolato semplicemente tramite i colori rosso/verde, facilmente confusi da persone affette da **daltonismo** di tipo deuteranopia e protanopia. Abbiamo inoltre inserito degli **alt nelle immagini** per una più facile lettura da parte degli screen reader.
- **TEORIA DEI COLORI**
  I colori utilizzati richiamano il logo del nostro sito, dando quindi un'idea di **continuità** e **contrasto** nelle varie parti della pagine, grazie alla **combinazione analoga** di giallo, arancione e rosa. Quelle che sono le parti essenziali alla navigazione, come la **navbar,** si riescono a distinguere perfettamente dal resto, in quanto rimangono sui colori del nero.
  Le **risposte** corrette/sbagliate sono evidenziate con i colori corrispondenti verde/rosso, dando un elevato contrasto e spiegando a colpo d'occhio l'esito ottenuto.
  Lo **sfondo** è stato interamente realizzato da noi, combinando icone inerenti alle varie categorie del gioco e offrendo un tocco di colore aggiuntivo al nostro sito.
- **VISUALIZZAZIONE DATI**
  Per rendere più chiara la visualizzazione delle **statistiche** dell'utente abbiamo utilizzato numeri con una certa formattazione: questi appaiono molto grandi e subito nella home, aiutando l'utente a cogliere nell'immediato le informazioni fornite dagli stessi. Inoltre l'utilizzo di una **barra percentuale** in un colore sgargiante rende chiaro il valore della percentuale di successi sul totale delle partite giocate.

Sicurezza

Per quanto riguarda la sicurezza abbiamo preso in considerazione le criticità derivanti dall'inserimento di **autenticazione/registrazione** e dall'utilizzo di **input utenti** nel sito.
Per quanto riguarda la **memorizzazione delle password** e la **creazione/memorizzazione di token** abbiamo scelto di affidarci al servizio **clerk,** che offre una suite completa di strumenti per l'autenticazione e registrazione di utenti all'interno di un database esterno. In questo modo i dati rimangono nascosti a coloro che potrebbero voler eseguire attacchi verso gli utenti.
Il nostro sito presenta poi un input utente per la ricerca di altri giocatori; questo può essere soggetto a attacchi come **Cross-Site Scripting** (iniettare script dannosi in pagine web) o **SQL injection** (inserire codice SQL dannoso nei campi di input). Informandoci abbiamo scoperto che attacchi di Cross-Site Scripting potevano essere evitati tramite **DOMPurify**, abbiamo quindi deciso di inserirlo nella pagina interessata.
Inoltre per evitare SQL injection ci siamo avvalsi di **query parametrizzate** (ossia che utilizzano dei parametri sostituiti successivamente con i valori reali solo dopo che la query è stata compilata).

Documentazione

- BOOTSTRAP (https://getbootstrap.com/docs/5.3/getting-started/introduction/);
- CSS (https://developer.mozilla.org/en-US/docs/Web/CSS);
- W3SCHOOLS (https://www.w3schools.com/cssref/index.php);
- NPM (https://docs.npmjs.com/);
- TYPESCRIPT (https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html);
- CLERK (https://clerk.com/docs);
- MATERIALE DIDATTICO.

Prossimi passi...

In quanto il nostro sito si presenta come una **DEMO** iniziale del gioco, ci si propone di aggiungere nuove funzionalità in futuro e migliorare quelle già presenti.
Per migliorare il sito, ci si potrebbe concentrare su diversi aspetti:

- utilizzo del **session storage** del browser anche per gestire i dati temporanei che vengono persi in fase di disconnessione (per ovviare a questo abbiamo reso inaccessibile la navbar durante le partite ma l'utente ha comunque sempre accesso a quella del browser);
- ulteriori elementi di gamification, come **potenziamenti** e **livelli di difficoltà** per aumentare l'interattività e l'engagement degli utenti;
- aggiunta di una **chat**, per migliorare l'interazione tra utenti ed amici;
- fare in modo che durante le partite, le **domande** siano le stesse per ogni giocatore, in quanto ora sono scelte casualmente per ognuno (nonostante a noi non sembri essenziale);
- gestione del **tempo scaduto** nelle domande.