

Quiz App



Membrii echipei:

Micu-Honțan Valentin

Toader Ana Maria

Balagiu Darian

Proiectul a fost împărțit în mod egal, fiecare a luat parte la implementarea acestuia (atât pentru documentare, implementare și README).

Scurta descriere a aplicației implementate:

Aplicația implementează un site web ce permite unui utilizator creare de chestionare. Ca funcționalități, aceasta permite creare unui cont, navigarea între cele două pagini principale (cea de adăugare a întrebărilor și cea de vizualizare a chestionarului) și pagina de account details, ce conține datele introduse de utilizator la înregistrare. Într-un chestionar se pot adăuga un număr nelimitat de întrebări la care se poate răspunde din pagina "Dashboard". Fiecare utilizator are acces la propriile întrebări, iar o întrebare are 4 variante de răspuns.

Limbaje/Tehnologii folosite:

Backend: Aplicația utilizează *Flask*, *Python* și *SQLAlchemy*, pentru interacțiunea cu baza de date.

În continuare descriem scurt funcționalitățile implementate în fiecare fișier.

-În **main.py** se rulează aplicația

Prin linia: `if __name__ == '__main__':` -> ne asigurăm ca rulăm aplicația doar dacă se rulează fișierul, nu dacă se importă altundeva.

-În **init.py** se crează baza de date (dacă aceasta nu există deja) și se inițializează aplicația.

Aici am înregistrat și Blueprint-urile pe care le-am creat (views și auth).

Prin linia: `app.config['SECRET_KEY'] = 'very important'` -> se encipitează/securizează cookies și session data website-ului.

-În **views.py** am creat rutele aplicației

Aici am stocat rutele aplicației ce pot fi accesate de utilizator (pagina de login, home, dashboard, etc.).

Am folosit Blueprint pentru a structura aplicația prin organizarea în subdirectoare. În Flask, views primesc requests și returnează răspunsuri pentru acele requests. Orice obiect de tip Blueprint are decoratori (ex: route) ce permit asocierea unor operații ce vor fi executate în aplicație. Metodele definite aici vor randa (*render_template()*) fișierele .html corespunzătoare, definite în directorul *templates*.

-În **models.py** am creat două database models: User și Quiz

User va avea ca primary key câmpul id de tip întreg, pentru a putea identifica în mod unic un utilizator în sistem. Acest id nu trebuie setat manual, va fi asignat automat în mod incremental la adăugarea unui nou utilizator în baza de date.

Quiz va avea, de asemenea, ca primary key un id întreg, identificând în mod unic chestionarele.

Pentru a modela relația dintre utilizator și chestionare folosim un foreign key. Astfel, un utilizator poate avea mai multe chestionare.

-În **auth.py** sunt definite acțiunile pentru paginile de login, signup și logout

Login și sign up acceptă requests de tip POST, prin care prelucrează informațiile furnizate de utilizator pentru a adăuga noi entități în baza de date (adică a crea noi conturi). Aici se verifică diverse criterii, precum lungimea minimă de 6 caractere a unei parole, potrivirea celor două parole furnizate la înregistrare, etc.

Frontend: HTML si Javascript.

Fișierul base.html este fișierul de bază, iar toate celelalte îl extind. În acesta este importat bootstrap și este încărcat fișierul js necesar pentru bootstrap.

În restul fișierelor html sunt setate stilurile, culoarea, fontul, dimensiunea, butoanele și text box-urile iar în dashboard.html sunt folosite și snippeturi de cod Python pentru a itera prin toate întrebările utilizatorului curent. De asemenea, în base.html este folosit Python pentru a verifica dacă un utilizator este autentificat, iar în funcție de rezultatul acestuia se vor afișa diferite butoane pe navbar.

Instrucțiuni de rulare:

- Instalați ultima versiune de Python.
- Deschide terminalul și executa următoarele comenzi:

```
pip install -r requirements. txt
```

- Descarca repository-ul sau arhiva proiectului
- Deschide un terminal in folderul de baza al proiectului. Apoi, scrie următoarea comanda:

```
python3 main.py
```

- Website-ul va rula pe: <http://127.0.0.1:5000/>

Screenshots:

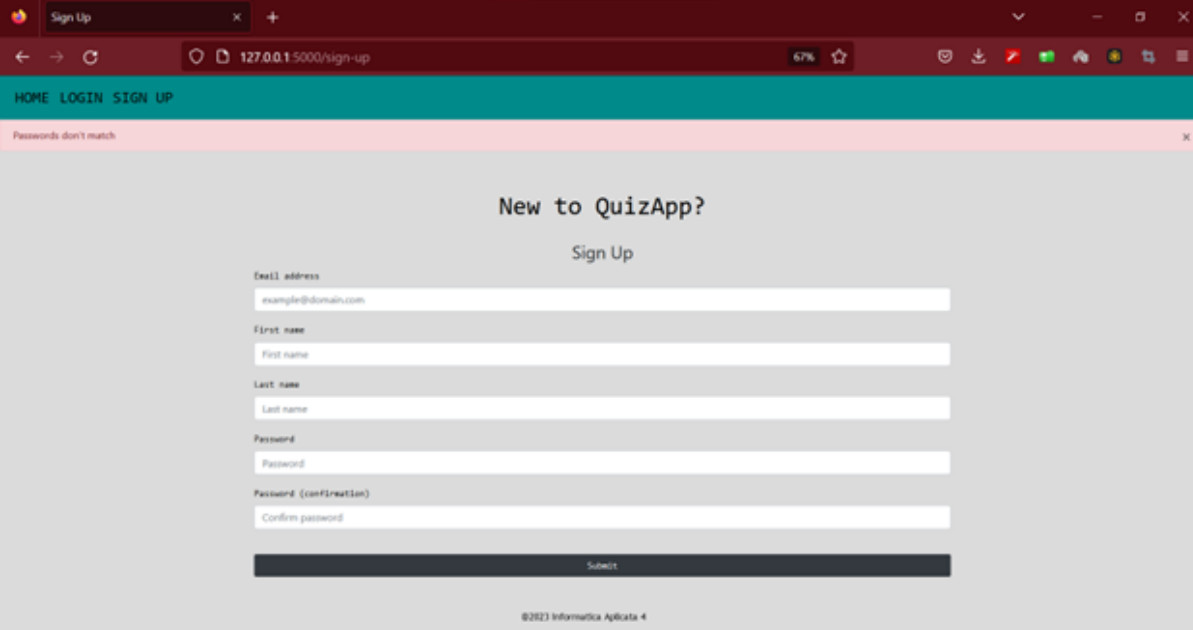
1. Pagina de log in.

The screenshot shows a web browser window with the title 'Login'. The address bar displays '127.0.0.1:5000/login?next=%2F'. The page has a teal header with links 'HOME', 'LOGIN', and 'SIGN UP'. Below the header, there are two green notification banners: 'Account created successfully!' and 'Please log in to access this page.' The main content area is titled 'Already have an account?' and 'Welcome back!'. It contains a login form with fields for 'Email address' (containing 'borin@upb.ro') and 'Password' (containing six asterisks). A 'Login' button is positioned below the password field. At the bottom of the page, the footer text reads '©2021 Informatica Aplicata 4'.

2. Pagina de Sign Up. Aici vor fi introduse date precum: adresa email, numele, prenumele și parola formată din minim 6 caractere.

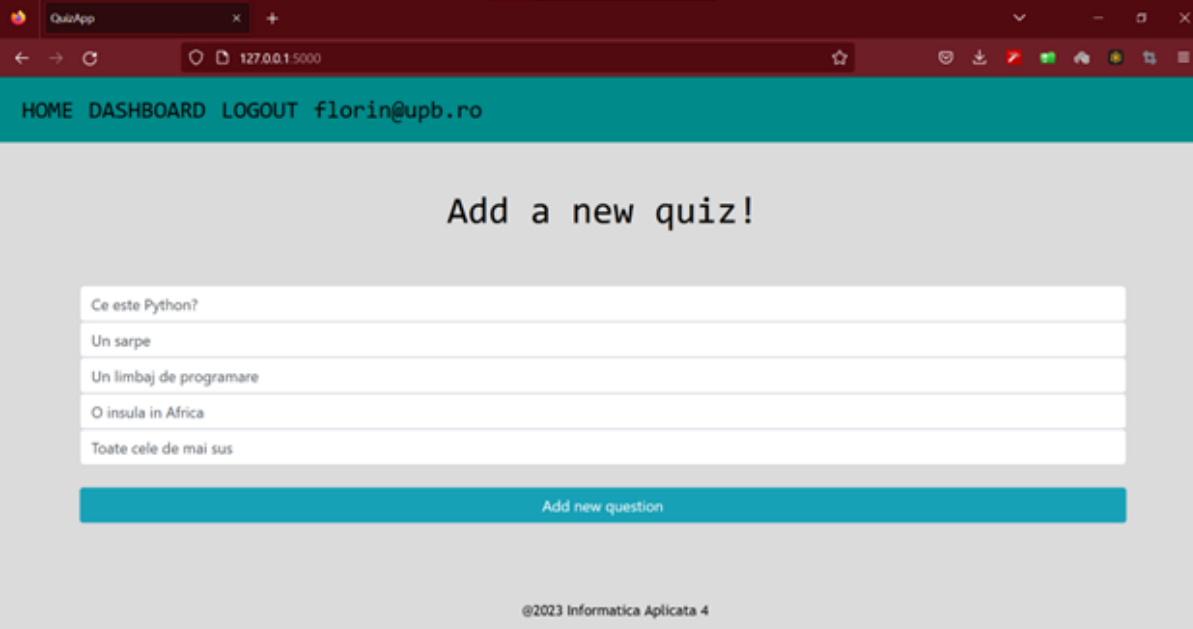
The screenshot shows a web browser window with the title 'Sign Up'. The address bar displays '127.0.0.1:5000/sign-up'. The page has a teal header with links 'HOME', 'LOGIN', and 'SIGN UP'. The main content area is titled 'New to QuizApp?' and 'Sign Up'. It contains a sign-up form with fields for 'Email address' (containing 'borin@upb.ro'), 'First name' (containing 'Prenume'), 'Last name' (containing 'Nume'), 'Password' (containing six asterisks), and 'Password (confirmation)' (containing ten asterisks). A 'Submit' button is positioned below the confirmation password field. At the bottom of the page, the footer text reads '©2021 Informatica Aplicata 4'.

3. Mesajul de eroare care apare pe pagina de Sign Up, în cazul în care parolele nu coincid, sau parola este sub 6 caractere.



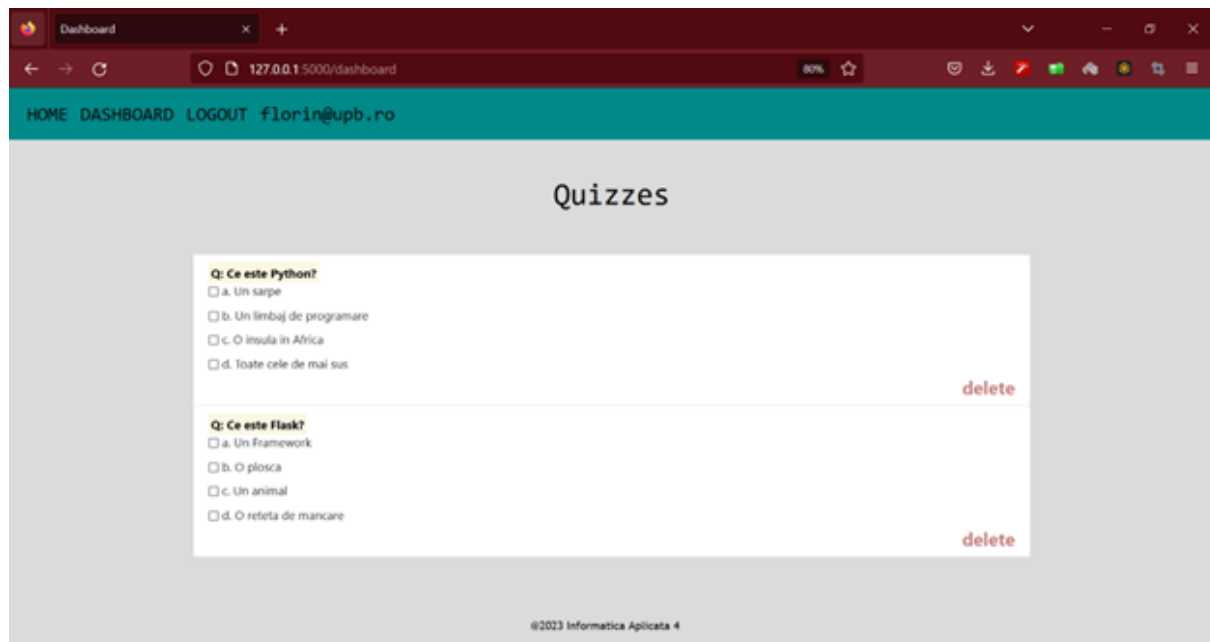
The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/sign-up". The page has a teal header with links "HOME", "LOGIN", and "SIGN UP". A red error message "Passwords don't match" is displayed at the top. The main content area is titled "New to QuizApp?" and "Sign Up". It contains a form with the following fields: "Email address" (with placeholder "example@domain.com"), "First name", "Last name", "Password", and "Password (confirmation)". A "Submit" button is at the bottom. The footer text is "@2023 Informatica Aplicata 4".

4. Pagina principala ("Home"). De aici, se pot adăuga oricâte întrebări pentru a crea un chestionar. În cazul în care răspunsurile nu sunt completate corespunzător (dacă textbox-ul este gol) se va afișa un mesaj de eroare.

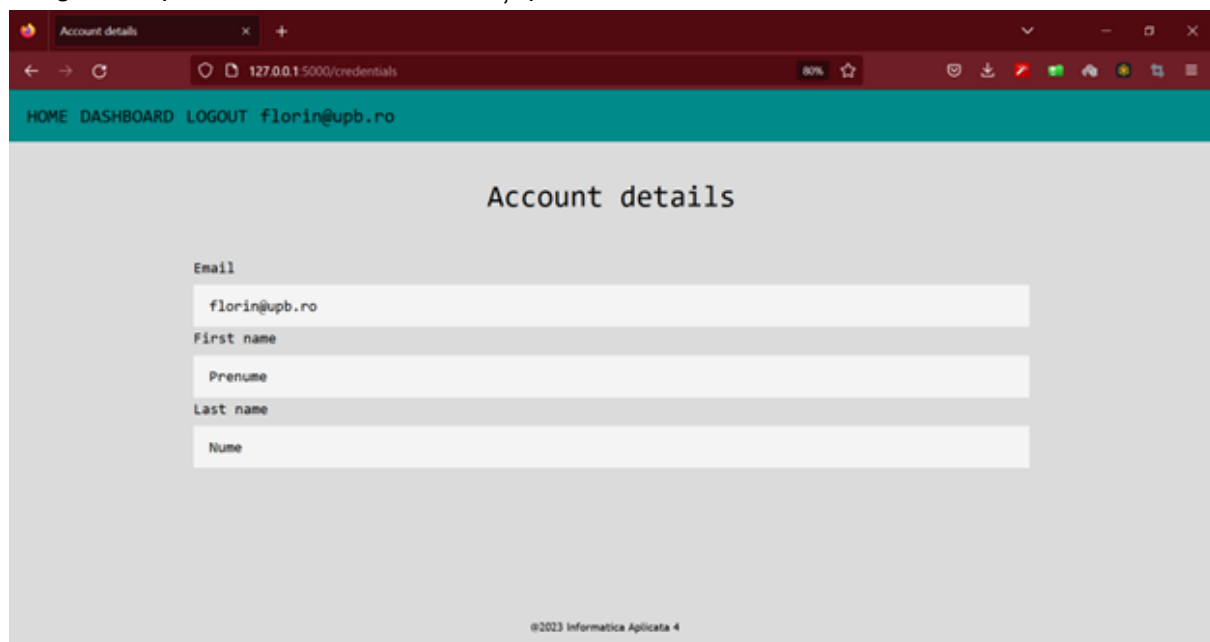


The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000". The page has a teal header with links "HOME", "DASHBOARD", and "LOGOUT", and the user email "florin@upb.ro". The main content area is titled "Add a new quiz!". It contains a form with five text input fields: "Ce este Python?", "Un sarpe", "Un limbaj de programare", "O insula in Africa", and "Toate cele de mai sus". A teal "Add new question" button is at the bottom. The footer text is "@2023 Informatica Aplicata 4".

5. Pagina de afișare a chestionarului, acționată după apăsarea butonului “Dashboard”. Aici se pot vizualiza toate întrebările, se pot șterge întrebări și răspunde cu răspuns unic sau multiplu la ele.



6. Ultima pagină, cea de privacy, deschisă în urma apăsării pe butonul cu email-ul propriu din navbar-ul website-ului. Aici se pot vizualiza datele despre cont furnizate de utilizator la înregistrare precum: email-ul, numele și prenumele.



Dificultăți întâmpinate:

Tema în sine a fost un exercițiu de documentare deoarece niciunul dintre noi nu avea cunoștințe în prealabil despre Python, Flask sau HTML decât minime. Inițial, am vrut să implementăm un quiz funcțional, în care utilizatorul putea încărca răspunsurile sale și să-și vadă scorul, să-l redea și eventual să fie și un leaderboard cu toți userii existenți, dar nu am reușit să facem tot ce ne-am propus.

Link către repo-ul proiectului pe github: <https://github.com/anatoad/QuizApp>.

Resurse:

- [1] <https://docs.python.org/3/>
- [2] <https://flask.palletsprojects.com/en/2.2.x/templating/>
- [3] <https://getbootstrap.com/docs/4.0/>
- [4] <https://www.youtube.com/watch?v=dam0GPOAvVI>
- [5] <https://flask-sqlalchemy.palletsprojects.com/en/3.0.x/>