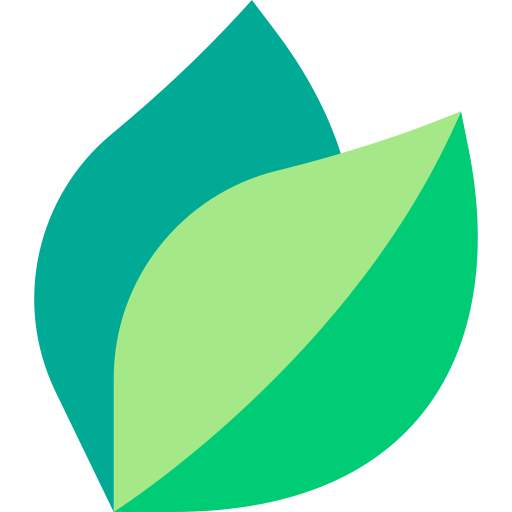
******

***Proiect I.S.W.***

TeaPython Application



***Manager*:** Olteanu Eduard Daniel

***Development*:** Momoi Alexandru, Olteanu Eduard Daniel

***Unit Testing*:** Momoi Alexandru

***Front-end Development / U.I.*:** Gheorge Luca-Teodor

***Documentație*:** Gheorghe Luca-Teodor,Teleki Ferenc-Tibor

***Database*:** Teleki Ferenc-Tibor, Olteanu Eduard Daniel

***Facultatea de Inginerie Electrică și Știința Calculatoarelor***

*Calculatoare Anul IV*

**Cuprins**

[1. Introducere 3](file:///C:\Users\Tibi\Downloads\Documentatie_AstroSwipe.docx#_Toc93085278)

[2. Front-end Mocks 3](file:///C:\Users\Tibi\Downloads\Documentatie_AstroSwipe.docx#_Toc93085279)

[Landing Page 3](file:///C:\Users\Tibi\Downloads\Documentatie_AstroSwipe.docx#_Toc93085280)

[Login Page 3](file:///C:\Users\Tibi\Downloads\Documentatie_AstroSwipe.docx#_Toc93085281)

[3. Librării Folosite 4](file:///C:\Users\Tibi\Downloads\Documentatie_AstroSwipe.docx#_Toc93085292)

[4. Arhitectura Aplicației 4](file:///C:\Users\Tibi\Downloads\Documentatie_AstroSwipe.docx#_Toc93085293)

[Metoda Management 4](file:///C:\Users\Tibi\Downloads\Documentatie_AstroSwipe.docx#_Toc93085280)

[Metoda Echipei 4](file:///C:\Users\Tibi\Downloads\Documentatie_AstroSwipe.docx#_Toc93085280)

[7. Development 21](file:///C:\Users\Tibi\Downloads\Documentatie_AstroSwipe.docx#_Toc93085294)

[8. Testare 22](file:///C:\Users\Tibi\Downloads\Documentatie_AstroSwipe.docx#_Toc93085295)

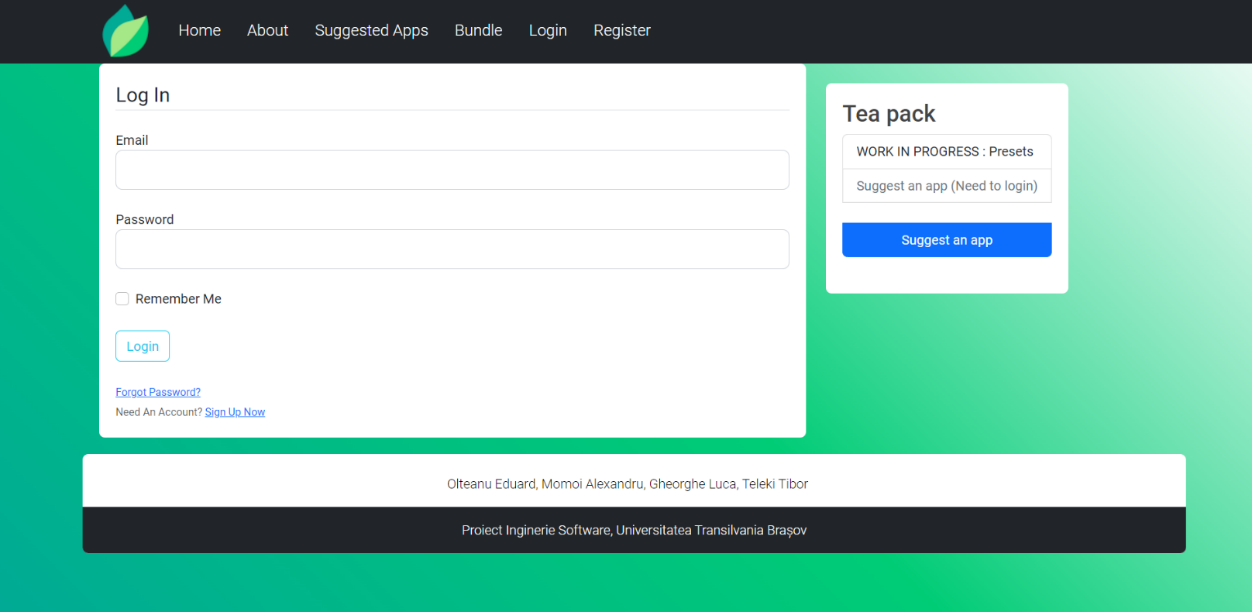
[9. Bibliografie 22](file:///C:\Users\Tibi\Downloads\Documentatie_AstroSwipe.docx#_Toc93085296)

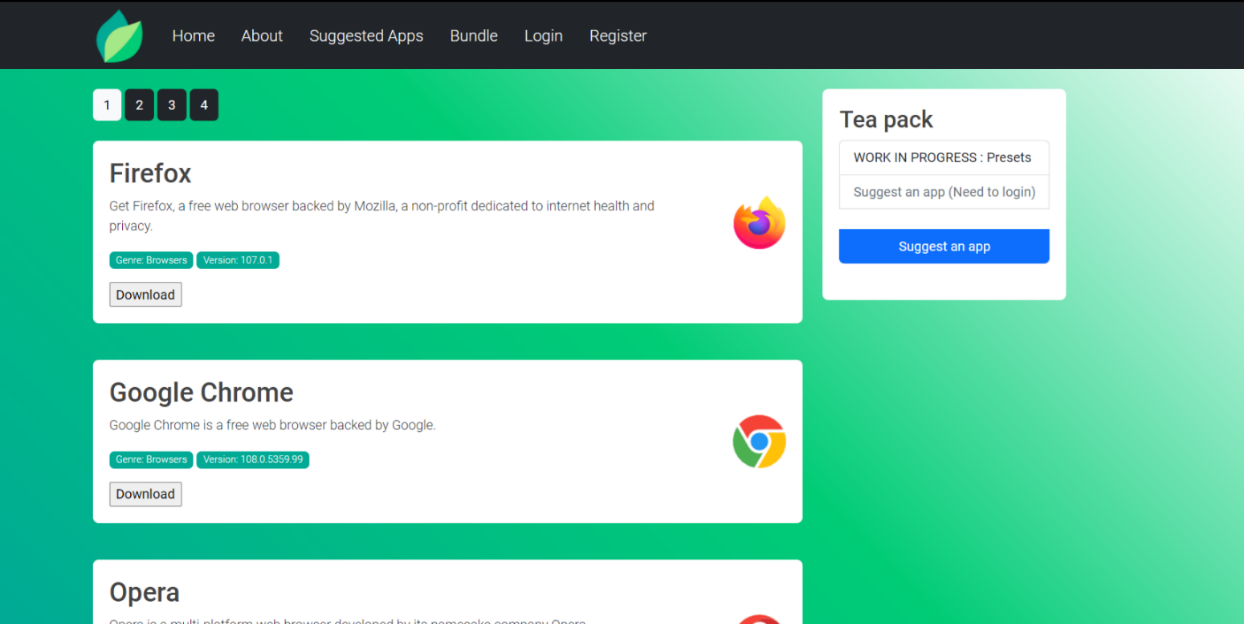
1. **Introducere:**

TeaPython este o aplicație web care folosește Flask API pentru a oferi utilizatorilor un AppStore unde își pot regăsi aplicațiile preferate. Aceasta îi oferă clientului posibilitatea de a descărca și instala mai multe aplicații simultan, dar poate opta și pentru bundle-uri prestabilite.

1. **Front-end mocks:**

* **Login Page și Landing Page**

****



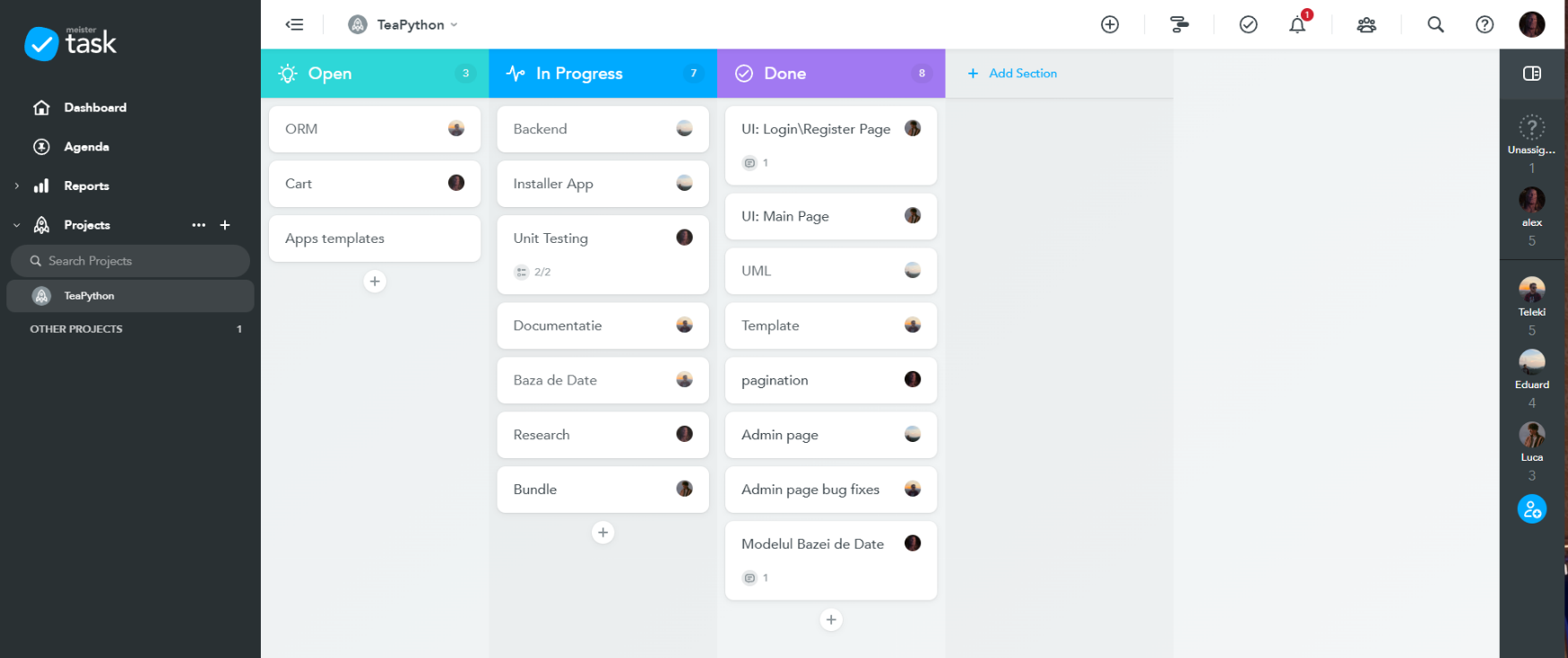
1. **Librării Folosite**

* Bcrypt-4.0.1 - Soluție pentru criptarea parolelor și a informațiilor
* Click-8.1.3 – Librărie în Python pentru generarea automată de pagini
* Dnspython-2.2.1 – Folosită pentru actualizări dinamice
* email-validator-1.3.0 – Librării pentru validarea mail-ului
* Flask-2.2.2 – Utilizată pentru dezvoltarea aplicațiilor web
* Flask-Bcrypt-1.0.1 – Extensie Flask
* Flask-Login-0.6.2 – Folositor pentru managementul user-ilor aplicației
* Flask-SQLAlchemy-3.0.2 – Suport pentru SQL Alchemy
* Flask-WTF-1.0.1 – Include încărcarea de fișiere
* greenlet-2.0.1
* idna-3.4
* importlib-metadata-5.0.0
* itsdangerous-2.1.2
* Jinja2-3.1.2
* MarkupSafe-2.1.1
* SQLAlchemy-1.4.44
* Werkzeug-2.2.2
* WTForms-3.0.1
* Zipp-3.10.0

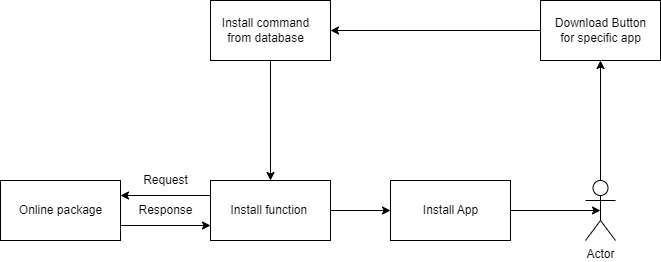
# **Arhitectura Aplicației**

* **Management-ul Echipei**

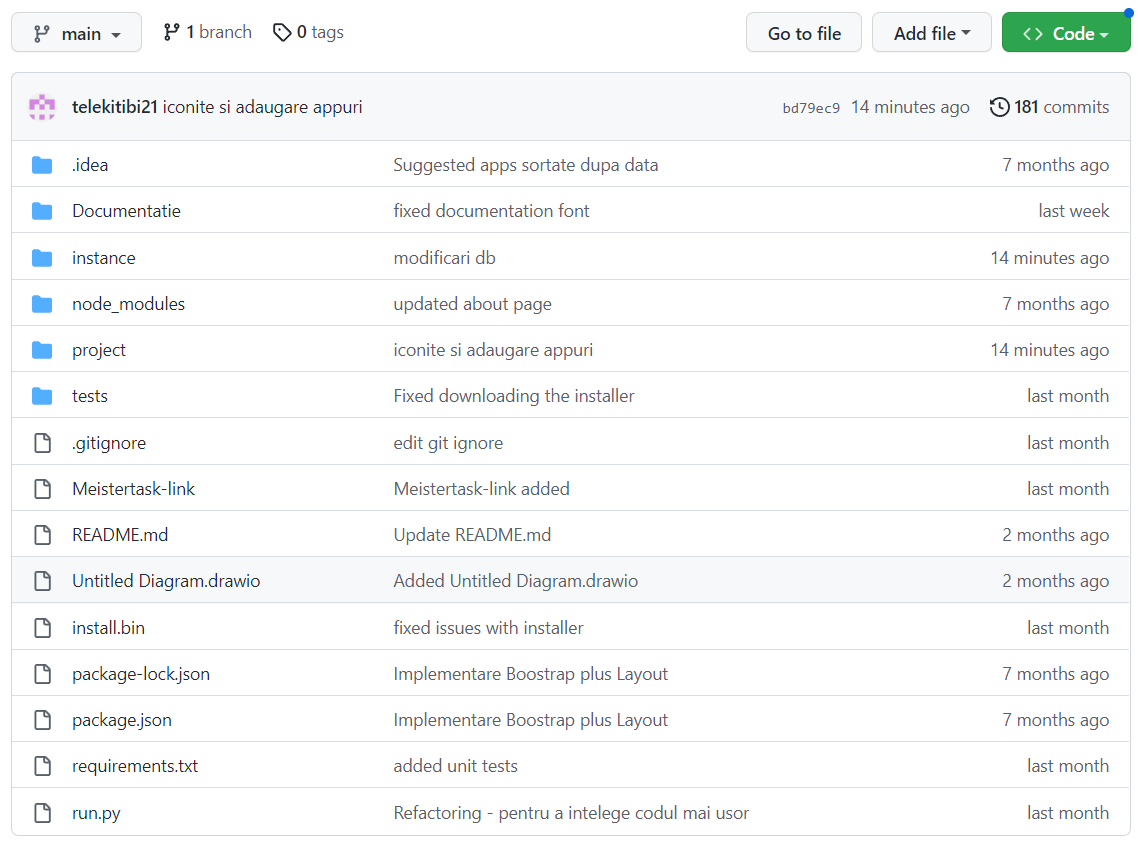
Pentru management-ul echipei am utilizat ca și board pentru task-uri, aplicația ***meistertask*.**



***Sursa:*** *1*

* **Diagramă Installer**
* **Versionarea codului**

Pentru versionarea codului am utilizat GitHub, mediu în care am creat branch-uri de dezvoltare pentru diferinte etape ale proiectului.



***Sursa:*** *2*

# **5. Management**

## Etape realizare:

* 1. Realizarea diagramelor U.M.L.
  2. Realizare mock-urilor pentru partea de front-end
  3. Implementarea segmentelor de bază (atât front-end cât si backend):
     + Sign-Up & Login;
     + Posibilitatea de a putea crea sugestii;
     + Editare cont;
     + Editare sugestii;
     + Installere.

## Cerințe funcționalitate:

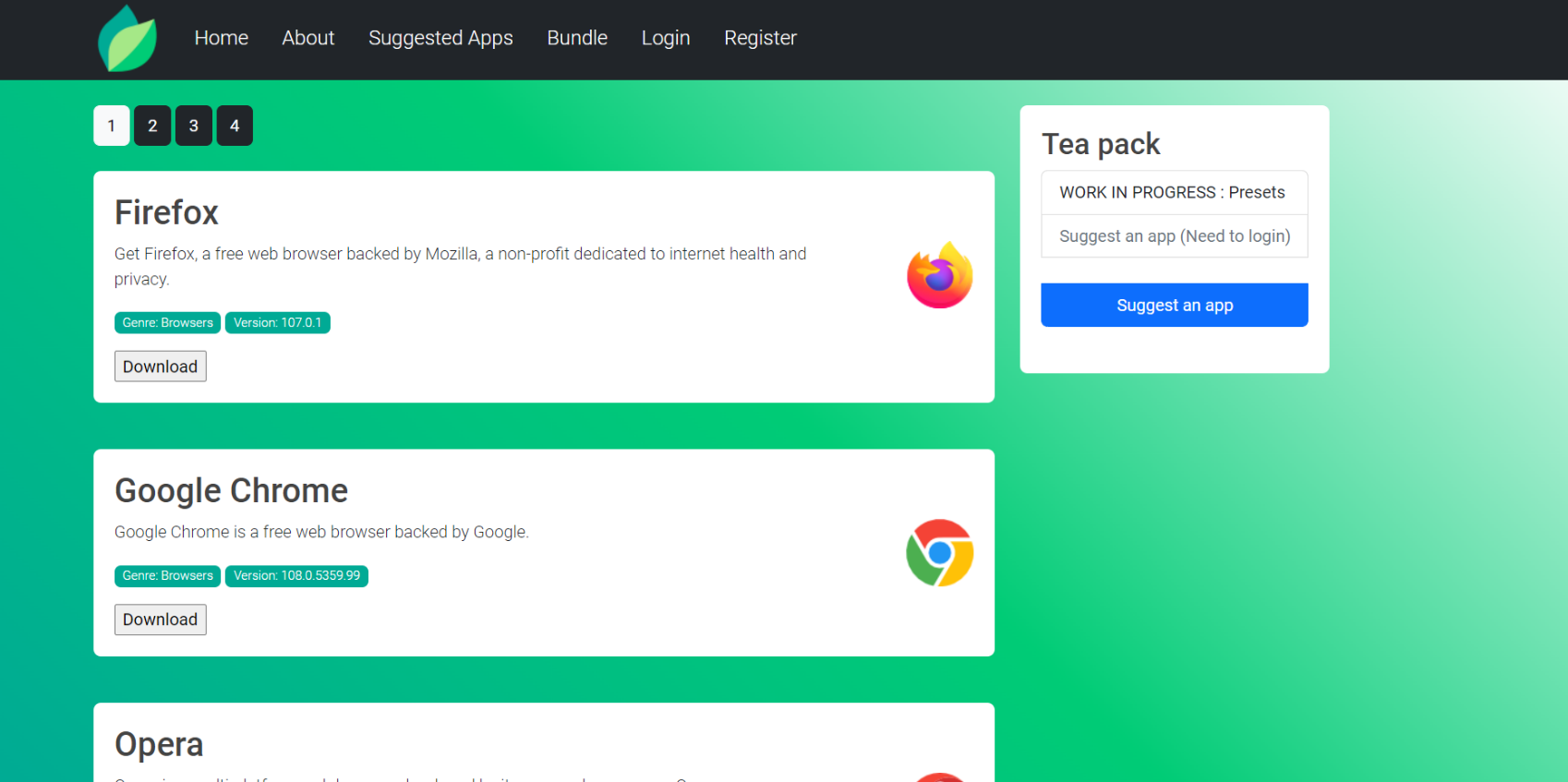
* + 1. Front-end:
       - Main Page: Login/Sign-up;
       - Home Page;
       - Aplicații Disponibile;
       - Profil.
    2. Back-end:
       - Salvare date utilizatori;
       - Crearea intaller-ului ***droit***;
       - Selectarea aplicațiilor;
       - Descărcarea aplicațiilor.

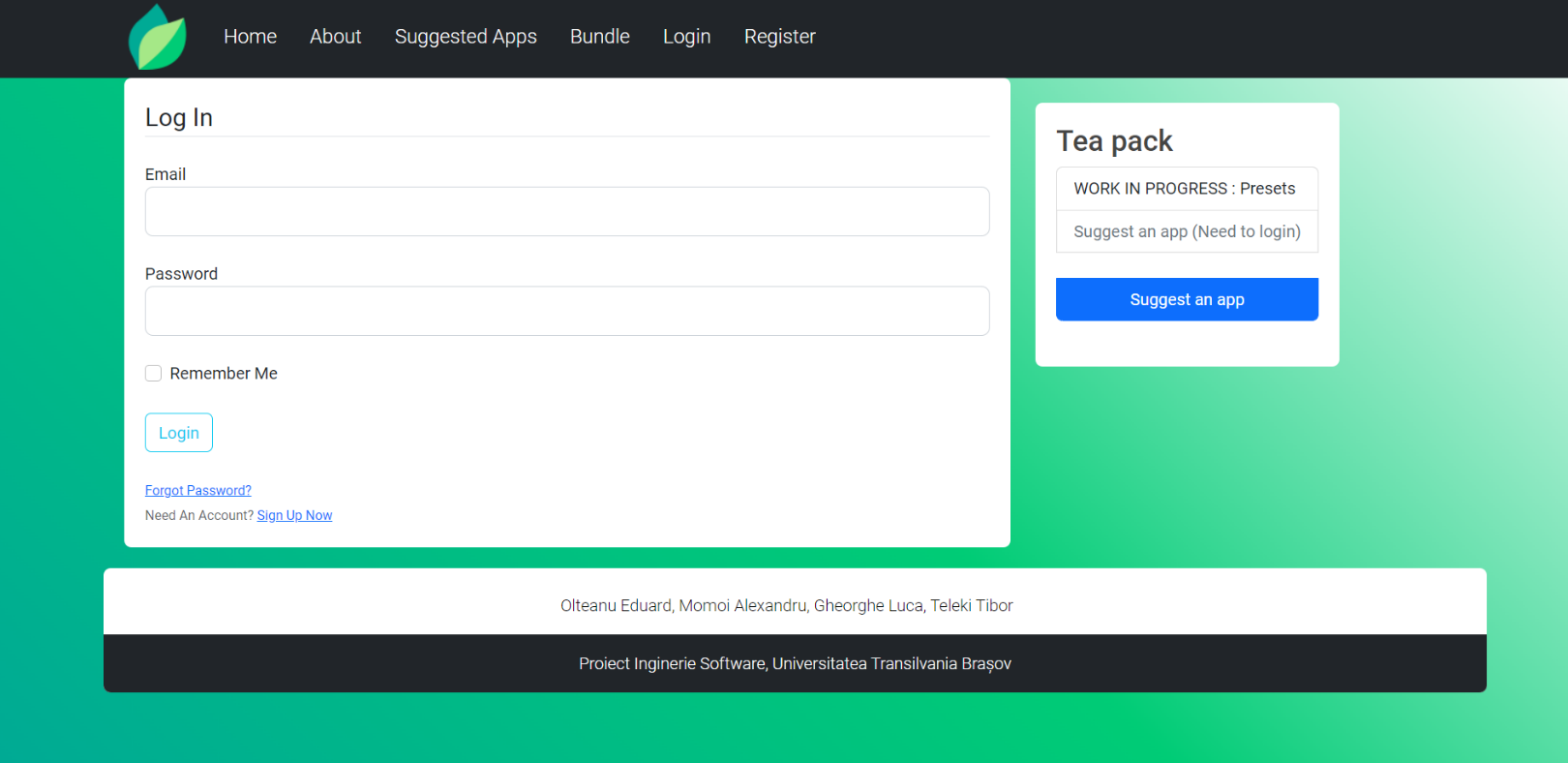
## Cerințe non-funcționale:

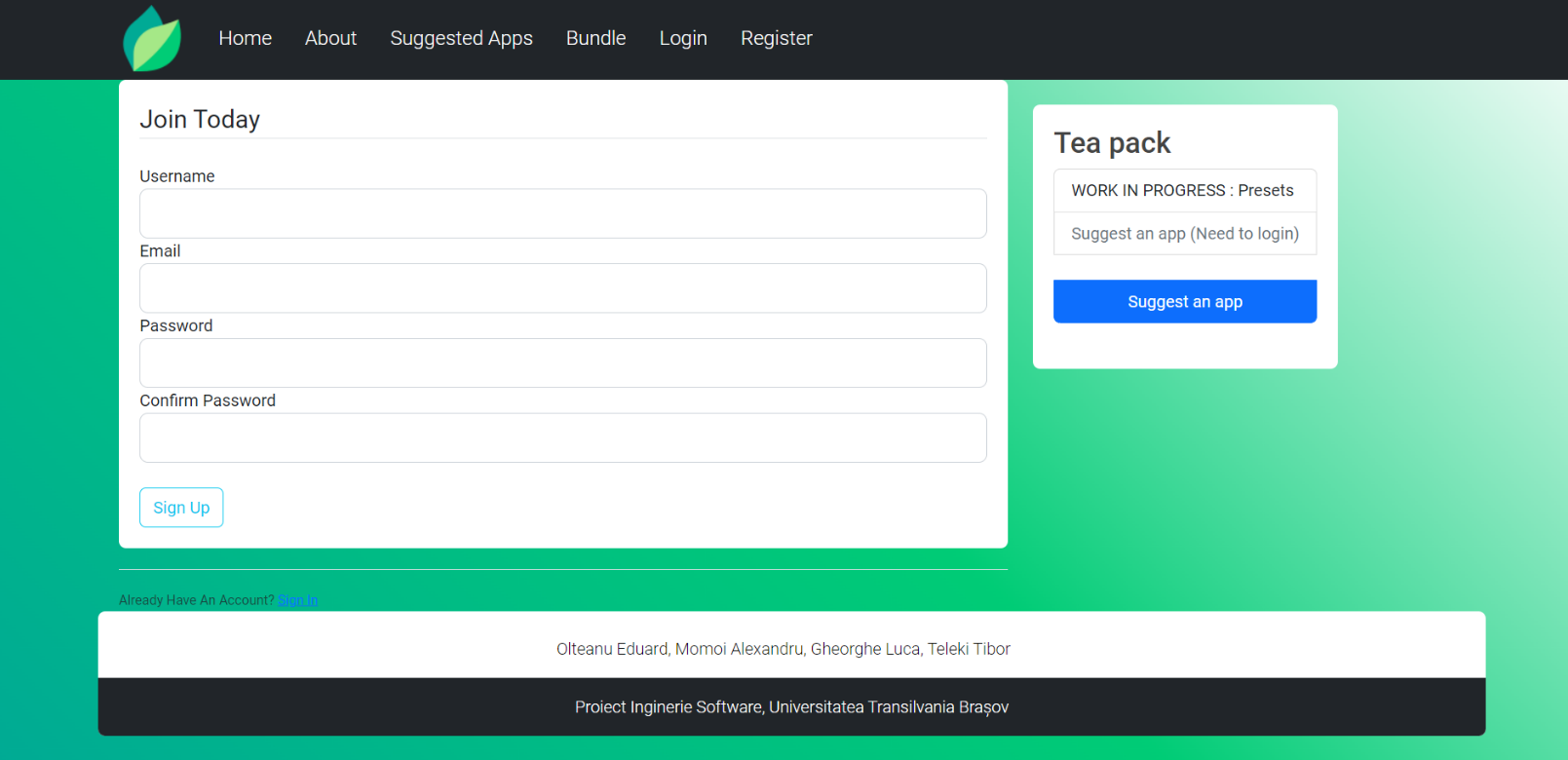
1. Front-end:
   * + - HTML;
       - CSS;
       - Python.
2. Back-end:
   * + - Python.
3. Database:
   * + - SqLite.

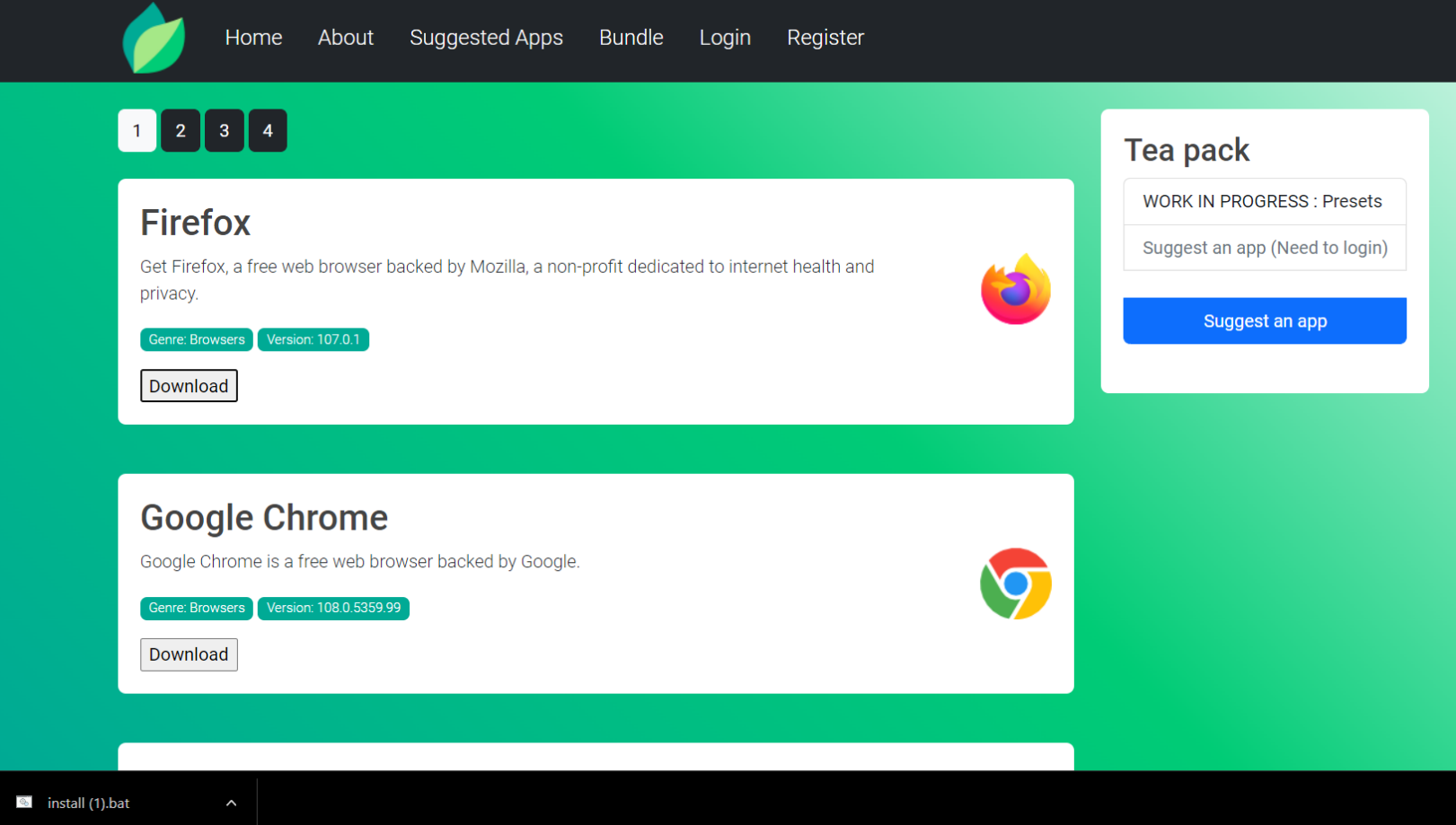
# **Tutorial**

*Utilizare*

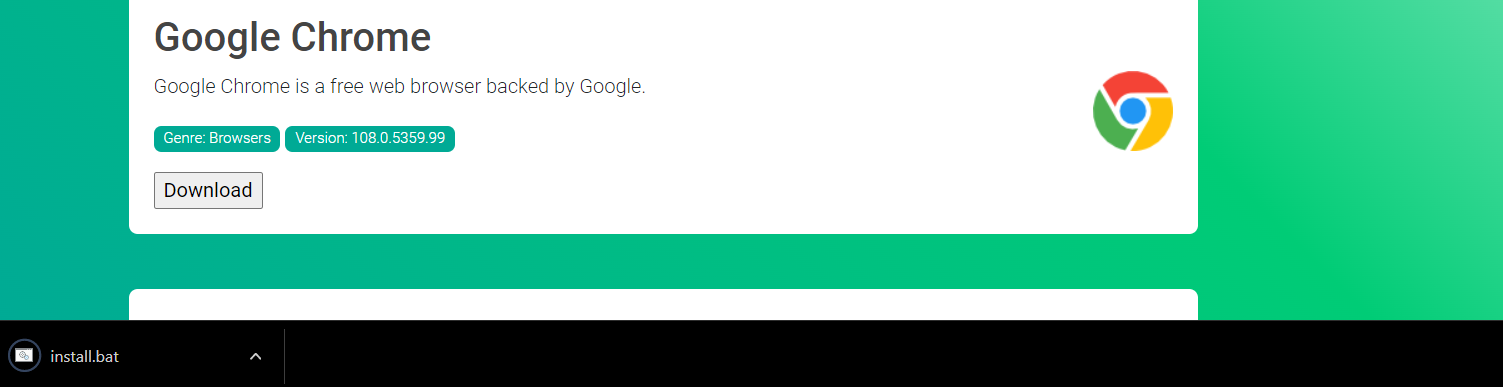
**Pasul 1:** Se deschide pagina web.

**Pasul 2:** În cazul în care ai cont, apeși pe butonul ‘*Login*’.

**Pasul 3:** În cazul în care nu există cont apasă pe butonul ‘*Register*’.

**Pasul 4:** Se alege aplicația dorită, se apasă pe download și va apărea fisierul ***.bat***.

**Pasul 5:** Se instalează aplicația dorită.



# 

*Instalare*

**Pasul 1:** Instalare folosind comanda ***pip.***

curl <https://bootstrap.pypa.io/get-pip.py> -o get-pip.py

python3 get-pip.py

sudo apt install python3.10-venv

**Pasul 2:** Se creează un "virtual environment” în folder-ul proiectului rulând comanda:

python3 -m venv teaenv

**Pasul 3:** Se activează “virtual environment-ul” cu pricina rulând comanda:

Source teaenv/bin/activate

**Pasul 4:** Se instalează cerințele:

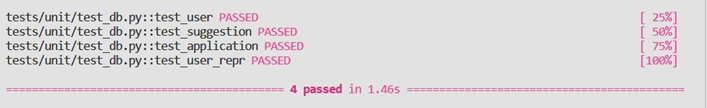
pip install -r requirements.txt

**Pasul 5:** Se rulează aplicația:

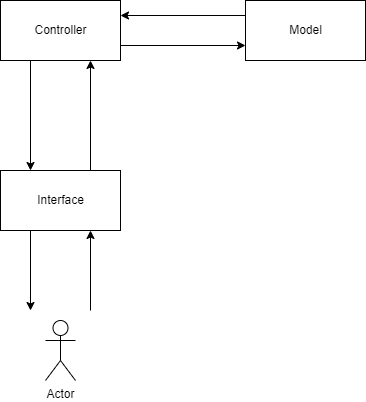
python3 run.py

Pentru rularea comenzilor de mai sus se va folosi Command Prompt.

# **7. Testare**



# **8. Development**



# **9.** **Bibliografie**

[1] <https://www.indeed.com/career-advice/career-development/waterfall-project-management>

[2] <https://www.meistertask.com/app/project/1FzFAJpz/teascript>

[3] <https://github.com/Eduard2609/TeaPython_ISW>