README - Human Benchmark

Description

Human Benchmark is a platform that offers a collection of simple, interactive games designed to measure your cognitive and physical abilities. Each activity allows individuals to test different skills, such as memory, reaction time, coordination, perception.

Programming & Front-End Development Languages

- *Python* - used for managing, collating and processing data
- *JavaScript* - used for the game implementations
- *HTML* - used for styling and base structure of the website
- *CSS* - used for styling and graphic details / interactions among the elements of the page

Running Instructions

- From project root execute the following command to activate the environment: `*.\.venv\Scripts\activate.bat*`
- After env activation, install flask with the following command (only the first time): `*pip3 install Flask Flask-SQLAlchemy mysqlclient flask-login*`
- To run the application, execute the following command: `*python -m flask --app app run*`
- To make it visible to others in the same network, add `*--host=0.0.0.0*`
- To run it in debug mode, add `*--debug*`
- Alternatively, implementation now supports running directly from VSCode, or with the use of the Makefile:
    - **make setup:** Download required dependencies
    - **make run-local:** Run only on the user machine
    - **make run-network:** Run and allow machines on the same network to access app
- run `*flask shell*` to access project shell

Contributions

Ailiesei Ana-Maria:

- ➔ created the Login Page and Login System using Flask
- ➔ created the Logout Page
- ➔ implemented the following games:
    - ◆ Reaction Time
    - ◆ Fast Typing

Dinca Alexandra:

- ➔ created the Home Page/User Page
- ➔ designed the basic template and style for all pages
- ➔ implemented the following games:
  - ◆ Sequence Memory Game
  - ◆ Pattern Memory Game
  - ◆ Quick Calculus Game

Georgescu Andrei:

- ➔ created the Sign-Up Page and Profile Page
- ➔ implemented database and associated I/O functions
- ➔ implemented base routing between pages

Savu Paul:

- ➔ created the Game Page
- ➔ created the routes for the 8 games
- ➔ implemented the following games:
  - ◆ Word Generation Game
  - ◆ Matching Colors Game
  - ◆ Aim Training Game

Difficulties

Ailiesei Ana-Maria:

- ➔ little experience with JavaScript/CSS/Python → followed tutorials and read documentation
- ➔ front-end and alignment of various elements → had to use scripts for aligning elements properly and come up with a few tricks for the best user visual experience
- ➔ never used Flask before→ learned how to create a user session and check credentials with the database, as well as create routes and understand app flow, page connection, and database manipulation

Dinca Alexandra:

- ➔ no previous experience with *JavaScript / CSS / HTML* → watched and read tutorials, followed documentation and relied on Bootstrap for certain template classes
- ➔ never worked on front-end and back-end at the same time → analyzed a few open source codes to understand syntax, coding style and logic and compromised by compressing both into one file (<style>, <html> and <script>)
- ➔ new to Flask → experimented on a separate branch how everything links together
- ➔ lacked game-design logic → broke down the back-end work-flow into smaller parts and created one at a time, until I was able to easily structure the algorithm into primitive variables, functions and the execution itself; as for the front-end, I struggled at first with animations until I learnt how to control and customize button actions

Georgescu Andrei:

- ➔ lack of previous experience with Python / CSS / HTML - watched tutorials and browsed Flask documentation.

- ➔ had to refresh prior knowledge of SQL, as well as its integration into Python through SQLAlchemy

Savu Paul:

- ➔ no previous experience with *JavaScript / CSS* → watched tutorials, read a few documentations and searched for question answers on various forum pages
- ➔ never worked with websites / Flask before → watched a tutorial covering basic aspects of this topic (creating routes, connection of pages, where / how code is written), followed the information provided in the university course

Git Repository:

https://github.com/AnduG/ProiectIA4