Универзитет „Св. Кирил и Методиј“ во Скопје
ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И
КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

Second midterm exam

HUMAN-COMPUTER INTERACTION DESIGN

03.06.2022

- Group 1 -

Task: Create a Django application for managing hot air balloon flights. Each flight is characterized by a mandatory code, name of the airport where it takes off and name of airport where it lands on, user who created the flight record, photo of the flight, information about the ballon, pilot and airline that realizes the flight. For each pilot, his name and surname, year of birth, total hours of flight and rank that he has in the company are kept. The type of balloon, the name of the manufacturer of the balloon and the maximum number of passengers in the balloon are kept for each balloon. For airline we kept its name, year of establishment and information on whether it flies outside Europe or not.

Additionally, information about the cooperation between airlines and pilots are kept. One pilot can be an associate of several airlines, and one airline can cooperate with multiple pilots.

Part 1:

Create the appropriate models to enable the system functionality.

It is necessary to enable the addition of objects through the Admin panel, with the note that the pilots-cooperators of an airline are added to the section for the airline. In addition, within the Admin panel you need to provide the following functionalities:

- When creating the flight, the user is assigned automatically according to the logged in user
- Once a flight has been defined and saved, it can only be changed by the user who created the flight
- Deletion of flights is not allowed for any user
- For pilots and airlines, the list shows only their names (and surname for the pilot)

Part 2:

Using the Bootstrap framework, create 2 templates to display system informations. Respectively:

- /index/ - display of general system information, shown in Image 1
- /flights/ - display of all flights created by the logged in user and departing from Skopje Airport. In this view, there is also a form for adding a new flight, shown in Image2

Once you have created the system, create a .zip file for entire project with all its accompanying elements. The database in the project remains the one defined by Django, SQLite. Create several objects from the models to test their functionality and let them stay in the database when attaching.

* Note: Only those applications that will compile successfully will be reviewed, ie the server will start successfully (if some functionality does not work for you, it is better to delete it).

Good luck!

working time: 120 minutes