Web Technologies Lab

LAB-08 **BSIT F20** June 7, 2023

The objective of this lab is to

- Practice Session and Cookies.
- Practice Python's web framework Flask.
- Perform insert and select queries on data via flask.
- Use the MVC approach for designing the solution.
- Perform data validation.
- Handle exceptions efficiently for handling unexpected scenarios.

Instructions!

- Keep your student identity cards with you.
- This is an individual lab, you are strictly **NOT** allowed to discuss your solutions with your fellow colleagues, even not allowed to ask how he/she is doing, as it may result in a negative marking.
- You can **ONLY** discuss this with TAs or Ma'am.
- Save your work frequently. Make a habit of pressing CTRL+S after every line of code you write.
- This is a **GRADED** lab, so, at the end of the lab session, you should have your complete work ready for evaluation.
- Follow proper coding conventions and write comments.
- ***Total Time for this Lab is 100 minutes.***

Task 40 Marks (100 Minutes)

Address Book Completion

For this lab, you will be completing Address Book of previous lab by adding/modifying some functionalities.

- **Login**
  - Develop a form-based web page where user will enter email and password to login to their account.
  - You should handle the input validations, and also properly report the invalid credentials entry to the user.
  - When a user login with the correct details, a session of that user will be maintained. User can see all contacts , Create , search contact.
  - When you login, all contacts will be shown on this Home page in the table.
  - Without login to the system, user can't access the contacts, either directly or from the URL.
- **Signup**
  - Develop a form-based web page where user will enter email and password information to sign up to your application.
  - Password should be greater than 8 characters, user appropriate form-fields, proper error handling identify unique email records validation.

- **Create Contact**
    - Provide form to get data in appropriate fields for new contact to be inserted.

- **Logout**
    - When a user logout, specific session will be destroyed and user will be re-directed to the login page.

- **Search Contact**

When Search the contact by entering contact number and it show proper message against the result.

In the database, you'll have the following tables:

- **Users**

  It represents all the users
    - **User_Id** <INT, pk, not null, unique, auto-increment>
    - **email** <VARCHAR, unique, not null>
        - *to represent the email of the user.*
    - **password** <VARCHAR, not null>
        - *to represent the password of the user.*
- **Contacts**

  It represents all the contacts
    - **Contact_Id** <INT not null, unique, auto-increment>
    - **User_Id** <INT, foreign nkey>
        - to represent the user that create the contact.
    - **name** <VARCHAR, not null>
        - *to represent the name*
    - **mobileno** <VARCHAR, not null>
        - *the mobile number*
    - **city** <VARCHAR, not null>
        - *the city of the user.*
    - **profession** <VARCHAR, not null>

  Composite key (user_id, mobile_no)

Grading Criteria

There are some things you've been doing repeatedly, and following them is mandatory, due to their level of being basic, those'll carry no marks, but not following them will result in a negative marking 😊.

Component	Requirement(s)	Marks
exceptions	Exceptions are handled carefully.	-2
proper names	The proper naming convention is used for variables, functions, classes, etc.	-2
MVC approach	Following the MVC approach, only file names are not going to do any better unless they are performing the functionality, they are meant to be,	1x3 = 3
app.py	Make sure to set the proper: ▪ Configuration File. ▪ Routes, rendering	5
templates	▪ Html file(s) are set properly, using the proper HTML elements, tags, and placeholders. (3) ▪ Data is displayed in the table with the proper header. (3)	6
Data validation	Data is validated and then stored in the database. ▪ Unique contact no. for each contact maintained. (3) ▪ No blank values are being dumped into the database. (1)	4
DB.py	Database class is well implemented and used.	3
Controller.py	A controller class is created and well-managed.	3
Flow	For full marks on this section, make sure to: ▪ add proper redirects (if multiple HTML pages are used) (5). ▪ give a better look to the HTML page. ▪ Overall flow is perfectly maintained across the components. ▪ Session Management (5)	6+5+5
Total		40

Programming is not about what out Know; it's about what you can figure out