COP4521 Programming Assignment 5: A Basic Flask Website

(Due: Nov 20, 2025, 11:59pm)

Objectives:

- Apply knowledge of relational database and SQL in practical scenarios.
- Develop a basic web application with a web frontend and a database backend for a small scale real-world applications using Flask and Python.

Description:

You can do this assignment in a group of two. In this assignment, you will use Python, SQLite3, and the Flask library to create a Flask website with the following pages:

- Home
- Add a New Baking Contest User
- List Baking Contest Users
- List Contest Results
- Results

The **Home** page should have three links

- Link - Add new Baking Contest User - (opens the add a New Baking Contest User page)
- Link - List Baking Contest Users - (opens the List Baking Contest Users page)
- Link - Baking Contest Results - (opens the list Contest Results page)

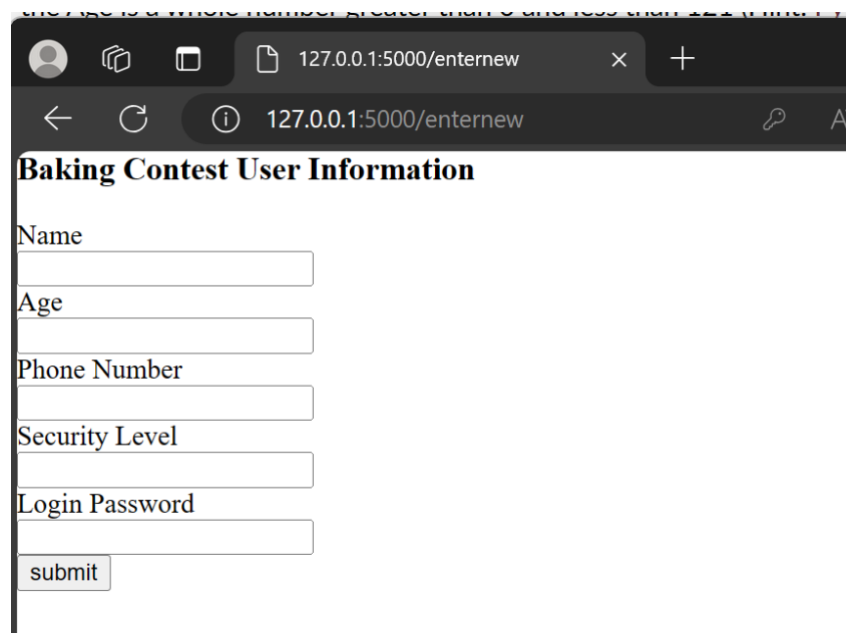The following is an example of the home page:
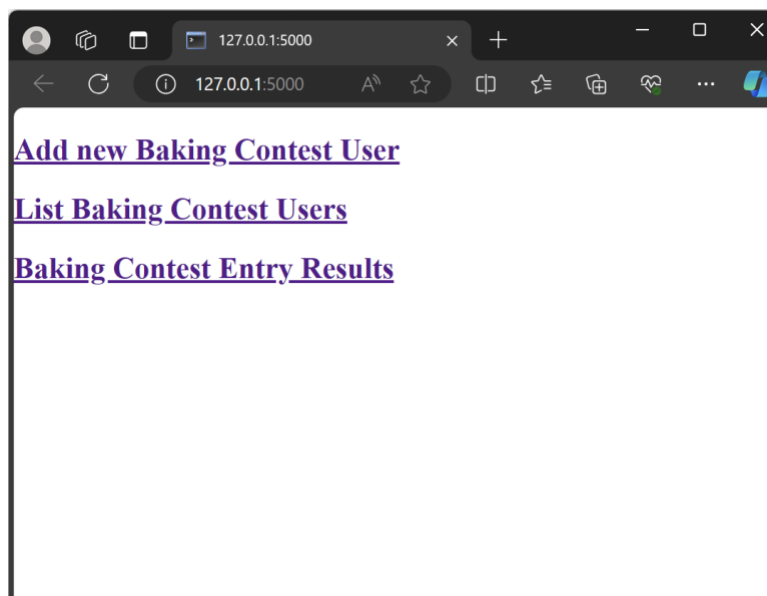
The **Add a New Baking Contest User** page should have

- a label and input text field for each possible attribute other than the id (Name, Age, Phone Number, Security Level, Login Password).
- a Submit button. When the Submit button is clicked: the values entered by the user should first be validated. If all inputs are valid then a record is added to the Baking Contest People table with the values entered by the user and a record added message is sent to the result page to display to the user. Otherwise, an error message is created indicating all input errors. This message is sent to the result page to display to the user.

  Input Validation Rules:

  - The Name is not empty and does not only contain spaces
  - The Age is a whole number greater than 0 and less than 121 (Hint: [Python - How to validate that a variable stores a numeric value](#))
  - The Phone Number is not empty and does not only contain spaces
  - The Security Level must be a numeric between 1 and 3. (Hint: [Python - How to validate that a variable stores a numeric value](#))
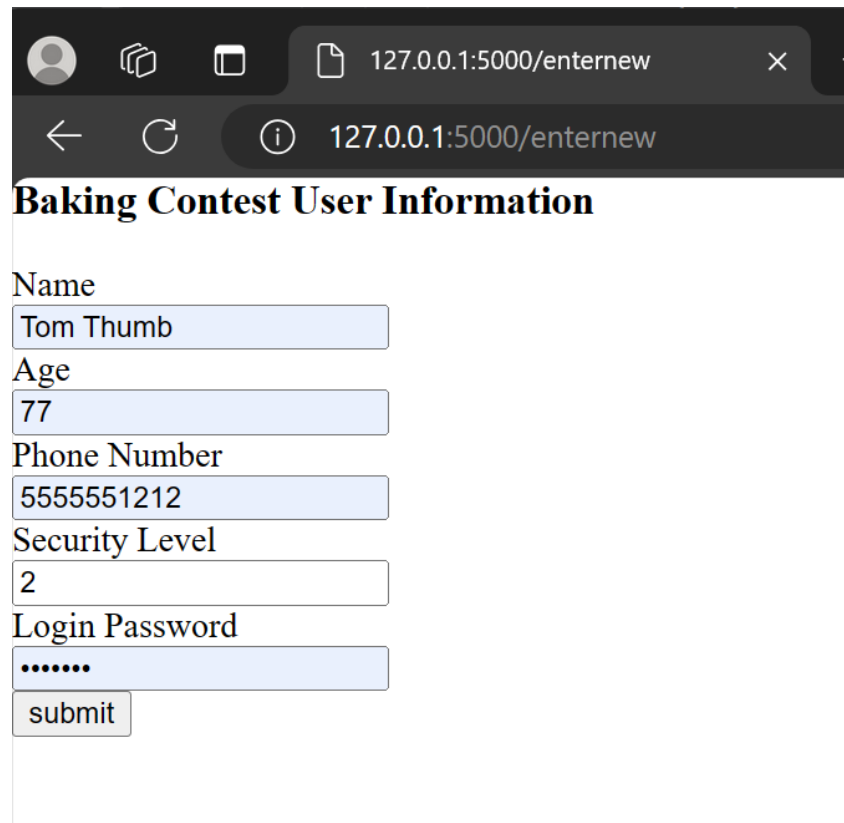 - The Login Password is not empty and does not only contain spaces

The following is an example of the Add a New Baking Contest User page (before any values are entered):

The following is an example of the page after values have been entered:
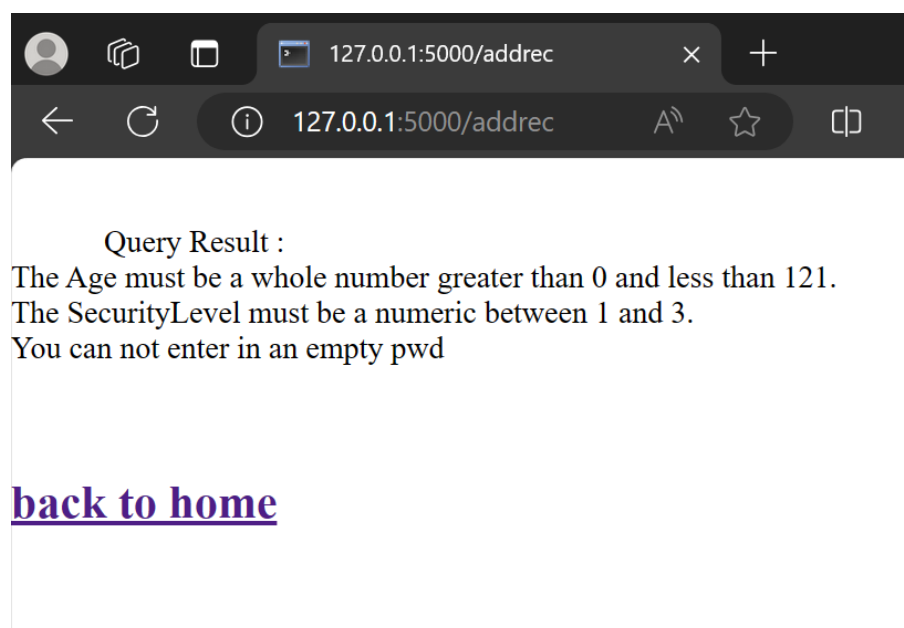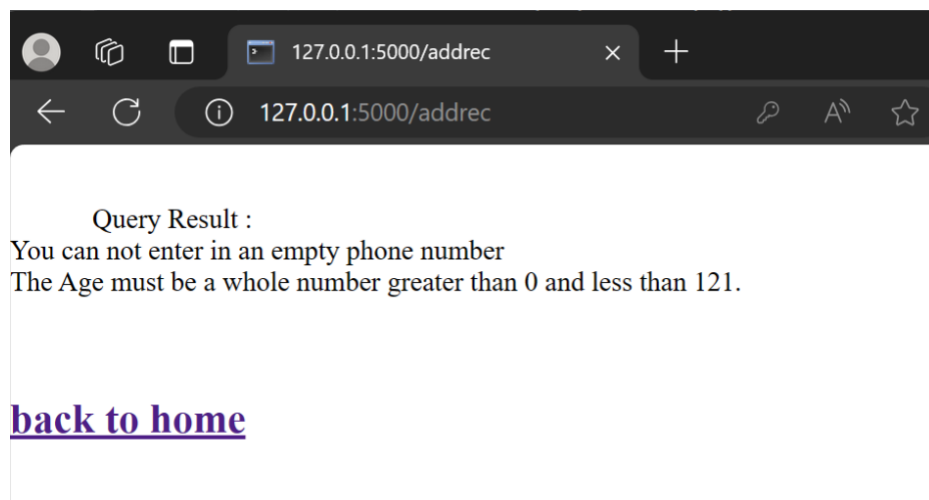

The following are examples of the different **results** pages after the user clicks the Submit button:

Success record added:
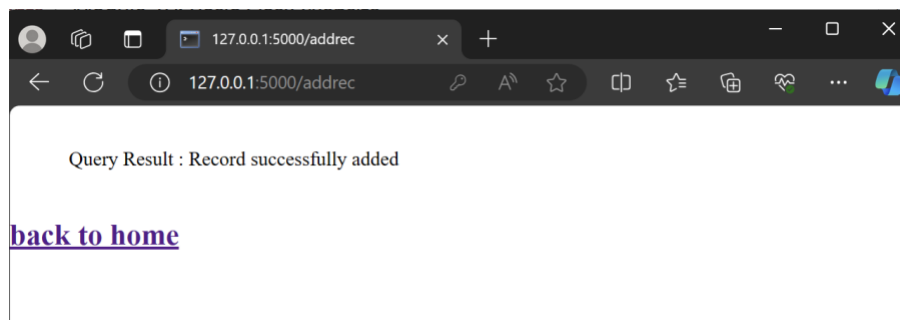
Various validation error messages:

Query Result :
You can not enter in an empty name
You can not enter in an empty phone number
The Age must be a whole number greater than 0 and less than 121.
The SecurityLevel must be a numeric between 1 and 3.
You can not enter in an empty pwd

back to home

Query Result :
You can not enter in an empty phone number
The Age must be a whole number greater than 0 and less than 121.

back to home

Query Result :
The Age must be a whole number greater than 0 and less than 121.
The SecurityLevel must be a numeric between 1 and 3.
You can not enter in an empty pwd

back to home

The **List Baking Contest Users** page should have

- a table that displays the following information for every record in the Baking Contest People table: Name, Age, Phone Number, Security Level, and Login Password

- Link - Go back to home page

The following is an example of the List Baking Contest Users page:

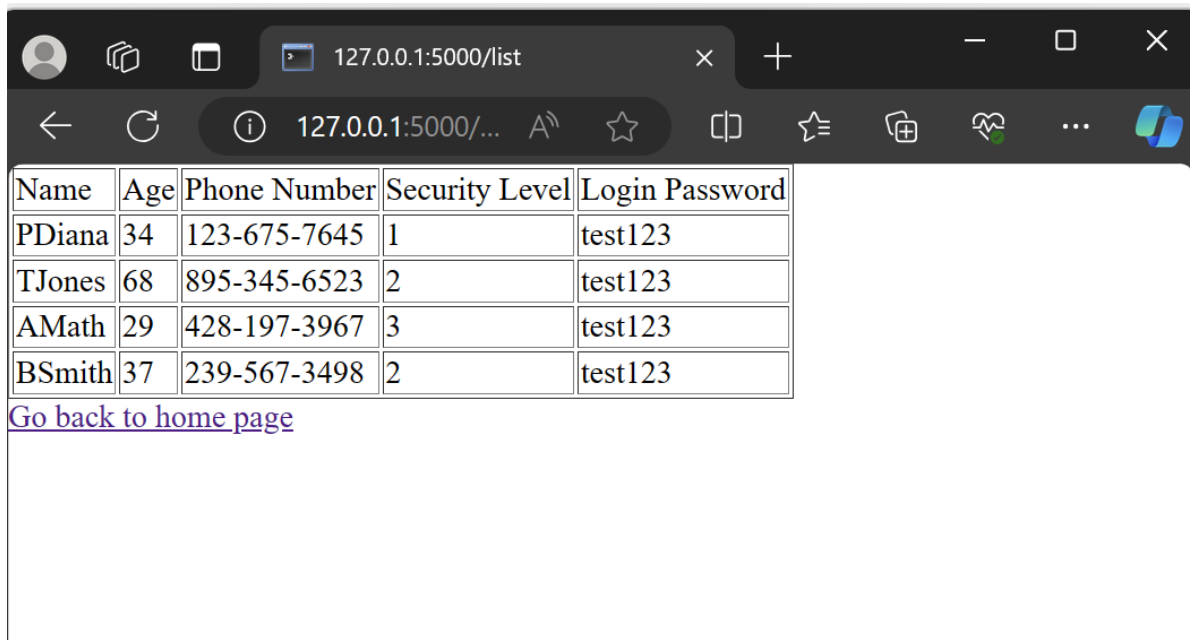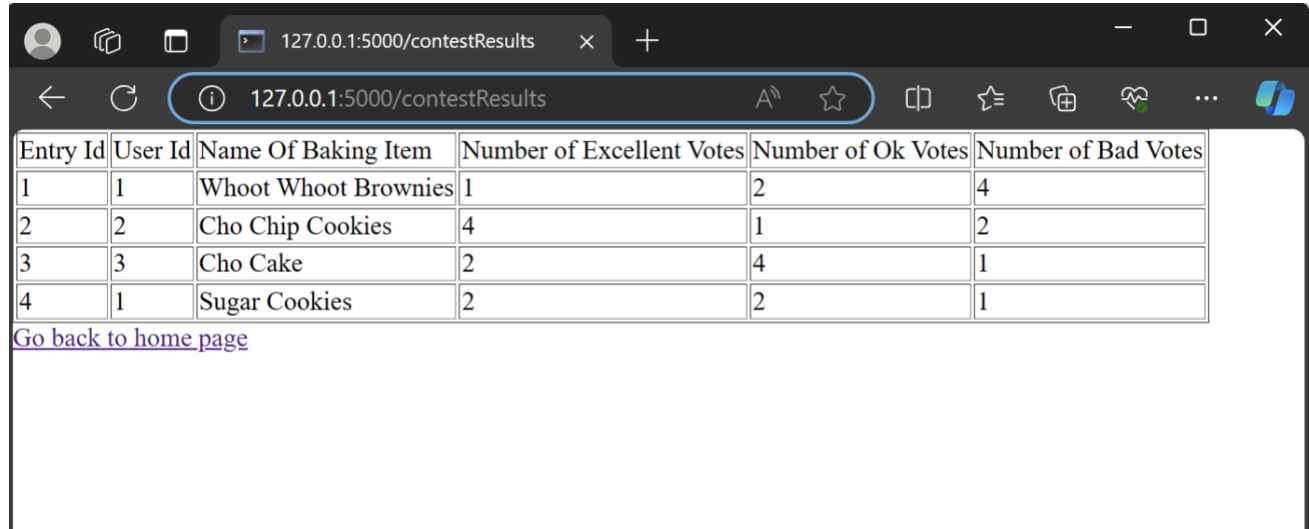| Name	Age	Phone Number	Security Level	Login Password
PDiana	34	123-675-7645	1	test123
TJones	68	895-345-6523	2	test123
AMath	29	428-197-3967	3	test123
BSmith	37	239-567-3498	2	test123

Go back to home page

The **List Contest Results** page should have

- a table that displays the following information for every record in the Baking Contest Entry table: Entry Id, User Id, Name Of Baking Item, Num Excellent Votes, Num Ok Votes, and Num Bad Votes

- Link - Go back to home page

The following is an example of the **List Contest Results** page:

Entry Id	User Id	Name Of Baking Item	Number of Excellent Votes	Number of Ok Votes	Number of Bad Votes
1	1	Whoot Whoot Brownies	1	2	4
2	2	Cho Chip Cookies	4	1	2
3	3	Cho Cake	2	4	1
4	1	Sugar Cookies	2	2	1

Go back to home page

The **Results** page should have

- display the value of the variable msg (Note this message could be a recorded added message or an input validation error record not added message)
- Link - Go back to home page.

Examples were given earlier.

Due date: Nov. 20, 2025, 11:59pm

Submission: Put all of your programs and files that are needed to run the website in a tar file. Name the tar file lastname_firstinitial_flaskwebsite.tar, and submit through canvas. Make sure that your website works on linprog before you submit your program.

Grading Policy:

- Include the basic header for assignments in a README.txt file in the root directory of your website (5 points)
- Home page functions correctly (10 points)
- Add a New Baking Contest User page functions correctly. (15 points)
- List Baking Contest Users page functions correctly. (15 points)
- List Contest Results page functions correctly. (15 points)
- Results page function correctly (10 points)

Note:

Make sure you develop and test thoroughly on linprog before you submit the files.

You will need to create initial SQL database tables for testing purposes.