

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

30 Marks (100 Minutes)

Address Book Completion

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

- **Delete Contact**
 - A webpage where all contacts will be displayed in a table form. The user should also be given an input text field, in which user will enter the name of the contact that user want to delete, clicking on the submit button will delete that contact and update the table accordingly.
Keep in mind two or more users can save the same contact. While deleting the contact make sure to delete the contact specific to that user.
- **Update**
 - When this option is selected, the user should be shown all the contacts in the addressbook table against the logged in user in a table form.
 - The user should also be given an input text field, in which user will enter the the name of the contact that user want to edit, clicking on the submit button will redirect user to a new route, where all the fields will contain the data of selected contact.

- User will update any field and as a result that record will be updated in the database accordingly.

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: <ul style="list-style-type: none"> ▪ Configuration File. ▪ Routes, rendering 	5
templates	<ul style="list-style-type: none"> ▪ 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. <ul style="list-style-type: none"> ▪ Unique name and contact 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: <ul style="list-style-type: none"> ▪ 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
Bonus Marks	Attractive User Interface using html/css/bootstrap	10
Total		50