Multi-Tier Programming (PROG43431)

Assignment 1

Due Date: See Slate

INSTRUCTIONS

- This assignment must be completed individually. Copying or reproducing the work done by others (in part or in full) or letting others copy or reproduce your own, and/or unauthorized collaboration will be treated as academic dishonesty under the College's Academic Dishonesty Policy.
- **IMPORTANT**: You must submit screenshot(s) demonstrating your work as instructed in the submission guideline. All screenshots MUST show your name and student Id (you can place a small text window containing your name and id on top of your screen, not hiding any content.) All screenshots must be readable in 100% zoom size. Your submission may not be marked if one or more required screenshot is missing or unreadable.
- Your application(s) must compile and run upon download to receive any mark. Your supplied outputs (e.g., screenshots) may not be marked, if the corresponding program fails to compile and run.
- To submit the assignments, follow the Submission Guideline provided at the end of this assignment.
- You must submit the assignment by the due date. The late submissions policy is specified in the Course Plan document available on Slate.
- Total marks = 60 (weight = 10% of the final grade).

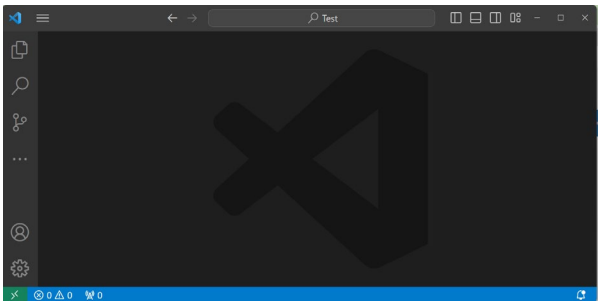
Part – I: Setting up the environment [marks: 10]

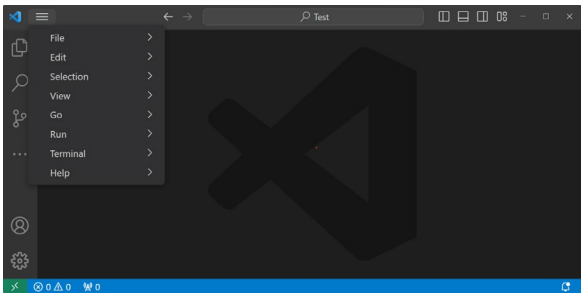
A. **Get VS Code IDE running on your machine**: We'll be using VS Code IDE in this course. If VS Code is not already installed in your system, please download, and install it from https://code.visualstudio.com/download. No customization will be required for the installation process - the default installation should be sufficient at the beginning. (though you will install several extensions later, as the course progresses). Here are LinkedIn Learning video demonstration links on installing VS Code:

   Windows: https://www.linkedin.com/learning/programming-foundations-fundamentals-3/installing-visual-studio-code-on-windows?resume=false&u=2272289

   Mac: https://www.linkedin.com/learning/programming-foundations-fundamentals-3/installing-visual-studio-code-on-a-mac?resume=false&u=2272289

   [Screenshot - 1] Run Visual Studio Code. Place a tiny text window containing your name and student id on the IDE. Take screenshots as below of the IDE running (including the text window on top) and paste it in the submission file.





B. **Familiarize with Installing Django**: We will be using Django as the application development framework and Python as the programming language. Install both Python and Django on your system. It is assumed from your previous Python courses that Python is already installed on your system and the environment path for Python is properly set in your system (if not, download it from https://www.python.org/ and install it.)

Use the following command in Windows command prompt to confirm if python is installed. If Python is properly installed the command will return the version number of Python installed on your system.

Use the following command in Windows command prompt to confirm if pip (preferred installer program) utility is installed on your system. If pip is properly installed the command will return the version number of pip installed on your system.

```
C:\Users\Syed Tanbeer>pip --version
pip 23.2.1 from C:\Users\Syed Tanbeer\AppData\Local\Programs\Python\Python311\Lib\site-packages\pip (python 3.11)
```

Installing Django:

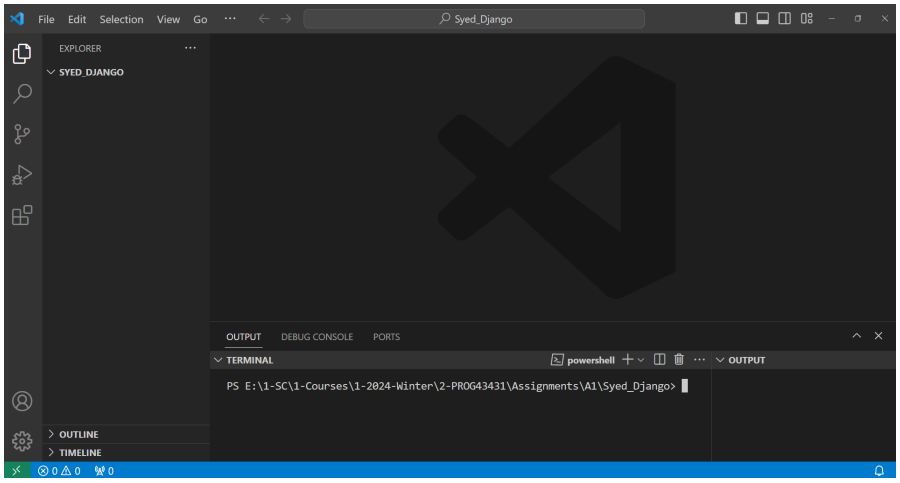
Follow the Django installation documentation at https://docs.djangoproject.com/en/5.0/topics/install/. Since Django includes a lightweight web server for testing applications, setting up Apache web server is not mandatory at this stage. The Apache web server will be required for hosting Django application on a production site, which you can skip at this point.

For Windows you can use this link: https://docs.djangoproject.com/en/5.0/howto/windows/ The following steps are based on Windows platform.

Though not mandatory, it is recommended that Django should be installed within a virtual environment. It will allow you develop Django projects with dedicated development environment (libraries, packages, etc.) with having its dependencies isolated from other Django projects.

- Step 1: Create a virtual environment named <YourFirstName_venv> follow the steps below:
  a) Create a folder named <YourFirstName_Django> using file explorer within your Assignment 1 solution folder.
  b) Open VS Code. Open the newly created folder in VS Code (File -> Open Folder…)

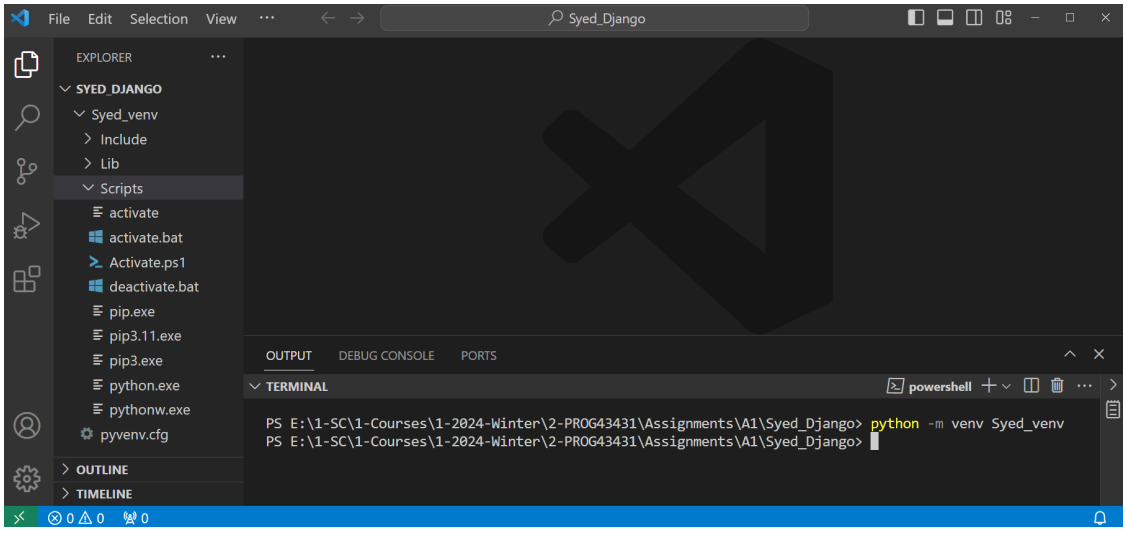
c) Open a new Terminal in VS Code (Terminal –> New Terminal). The terminal should show the path to your newly created folder.



d) Create a virtual environment using a name <YourFirstName_venv>. Use the following command in the terminal in VS Code, where *venv* is the module name to be used indicated by *-m* flag and '*Syed_venv*' is the name of the folder of virtual environment (replace 'Syed_venv' with your folder name):

python -m venv Syed_venv

It should create folder named <YourFirstName_venv> within the current directory i.e., <YourFirstName_Django>. Expand the folder, then expand the folder called Script in it as shown below.

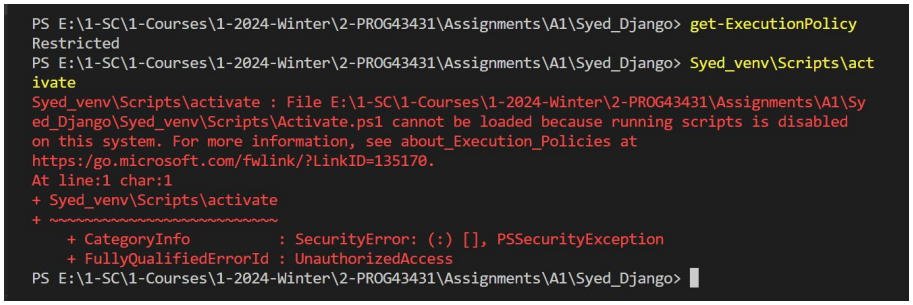


- Step 2: Activate the virtual environment, follow the steps below:

  Run the activate script available in the *Scripts* folder to activate the virtual environment using the command below (replace 'Syed_venv' with your folder name):

  > Syed_venv/Scripts/activate

If you receive following execution policy restrictions for the user, set the execution policy either 'Unrestricted' or 'Bypass'.

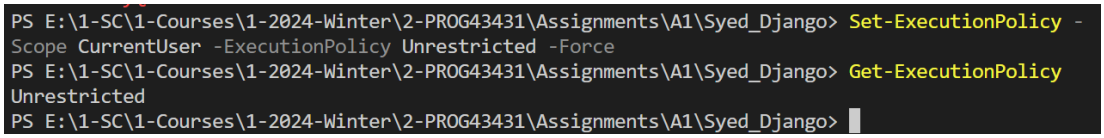


Use the following command to set it 'Unrestricted':

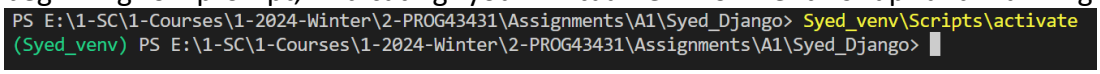
Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy Unrestricted -Force

To check if the execution policy is changed to unrestricted use command:

Get-ExecutionPolicy



Now use the virtual environment activation command again after fixing the execution policy. The name of your virtual environment should be shown at the beginning of prompt, indicating your virtual environment is up and running.

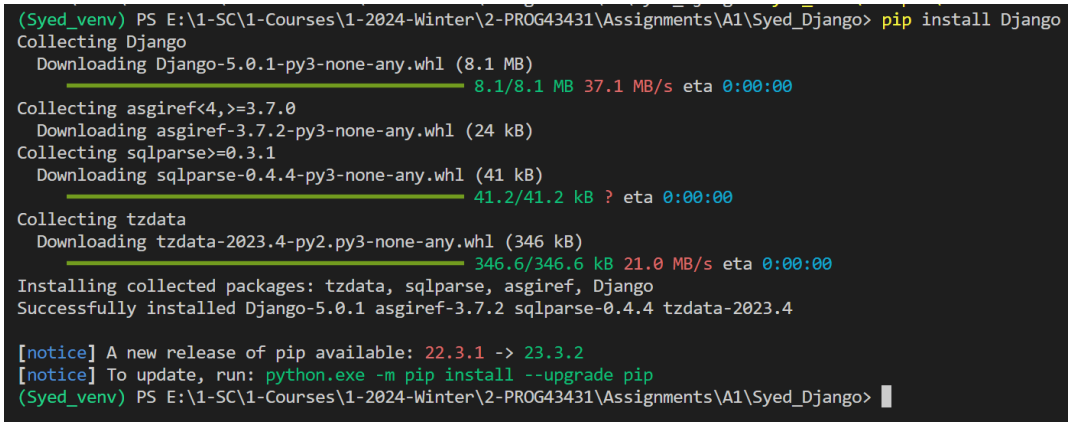


- Step 3: Install Django within the virtual environment:

  Use the following command to install Django:

  > *pip install Django*

  Following screenshot shows that Django is successfully installed.



To check the Django installation and version use the following command:

- Step 4: Create a project named <YourFirstName_project> in Django:

  Use the following *startproject* command to create a project within Django (replace '*Syed_project*' with your project name):

  > *Django-admin startproject Syed_project*

```
(Syed_venv) PS E:\1-SC\1-Courses\1-2024-Winter\2-PROG43431\Assignments\A1\Syed_Django> Django-admin startpro
ject Syed_project
(Syed_venv) PS E:\1-SC\1-Courses\1-2024-Winter\2-PROG43431\Assignments\A1\Syed_Django>
```

  The project should be created and listed in VS Code file explorer. Expand the project folder – locate the *manage.py* file in it. This is the main script file that you will use to work with the in-built server.

- Step 5: Run the project using the included web server:

  Navigate to the project folder (replace '*Syed_project*' with your project name).

  > *cd .\Syed_project*

  Then use the following command to start the server and run the project using the manage.py script.

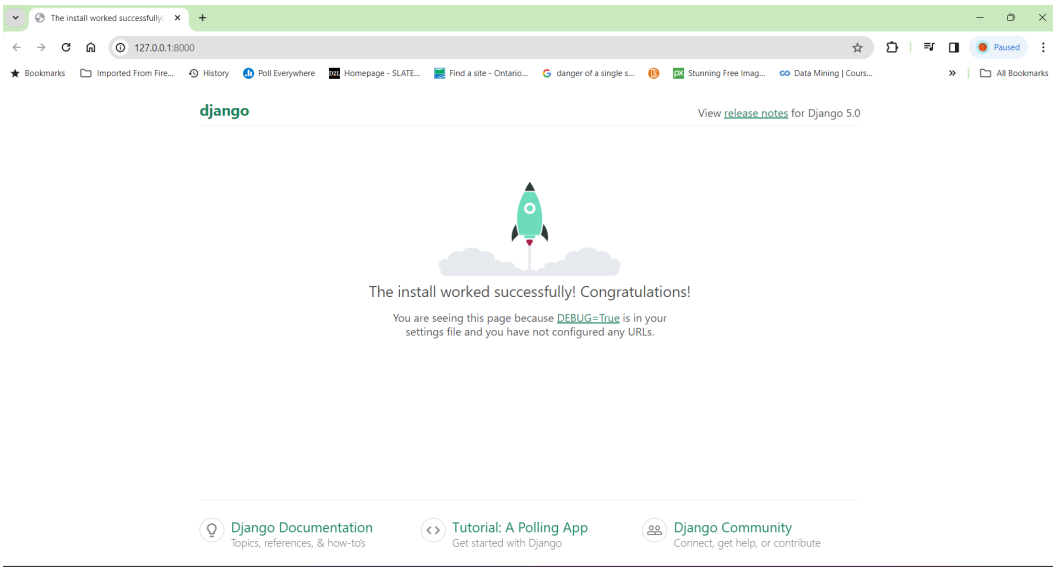
  > *python manage.py runserver*

```
(Syed_venv) PS E:\1-SC\1-Courses\1-2024-Winter\2-PROG43431\Assignments\A1\Syed_Django\Syed_project> python
 manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for
app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
January 09, 2024 - 20:22:32
Django version 5.0.1, using settings 'Syed_project.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

  Crtl+Click on the http://127.0.0.1:8000 url (or copy and paste it on browser address bar). The Django default page should render on your browser as below, indicating you have successfully installed Django, created a project, and are running it using Django server.



[**Screenshot -2**] Expand all the folder in your VS Code showing all files and subfolders in them. Place a tiny text window containing your name and student id on the explorer part o the IDE. Take screenshots (including the text window on top) showing all the files and folders in the explorer and paste it in the submission file.

[**Screenshot -3**] Render Django default page on browser. Place a tiny text window containing your name and student id on the page. Take screenshots (including the text window on top) of the page and paste it in the submission file.

Part – II: Working with HTML Forms [marks: 50]

Choose your form:

There are 10 different data entry forms listed in the table below (Table 1). You will be developing one of them using HTML, JavaScript and CSS. Select the form that corresponds to the last digit of your Sheridan student number. For example, if your student number ends with a 5, then you must develop a 'Grocery Item Order Form'. Again, students with Sheridan student number ending with a 0 must develop a 'Pizza Order Form'. Answer to a wrong form in full/part will not be marked and will get a zero.

Table 1 Data Input Forms Selection Table

Last digit of your student number	Your Form	Last digit of your student number	Your Form
0-1	Coffee Shop Ordering Form	2	Car Renting Form
3	Pizza Order Form	4	Sports Gear Order Form
5	Movie Ticket Order Form	6	Hotel book Order Form
7	Arts/Painting O-F	8-9	Book Order From
Opt	Forms for the WebSite you want to build (discuss your website idea with Rachel)		

If you decide to pick the Opt:

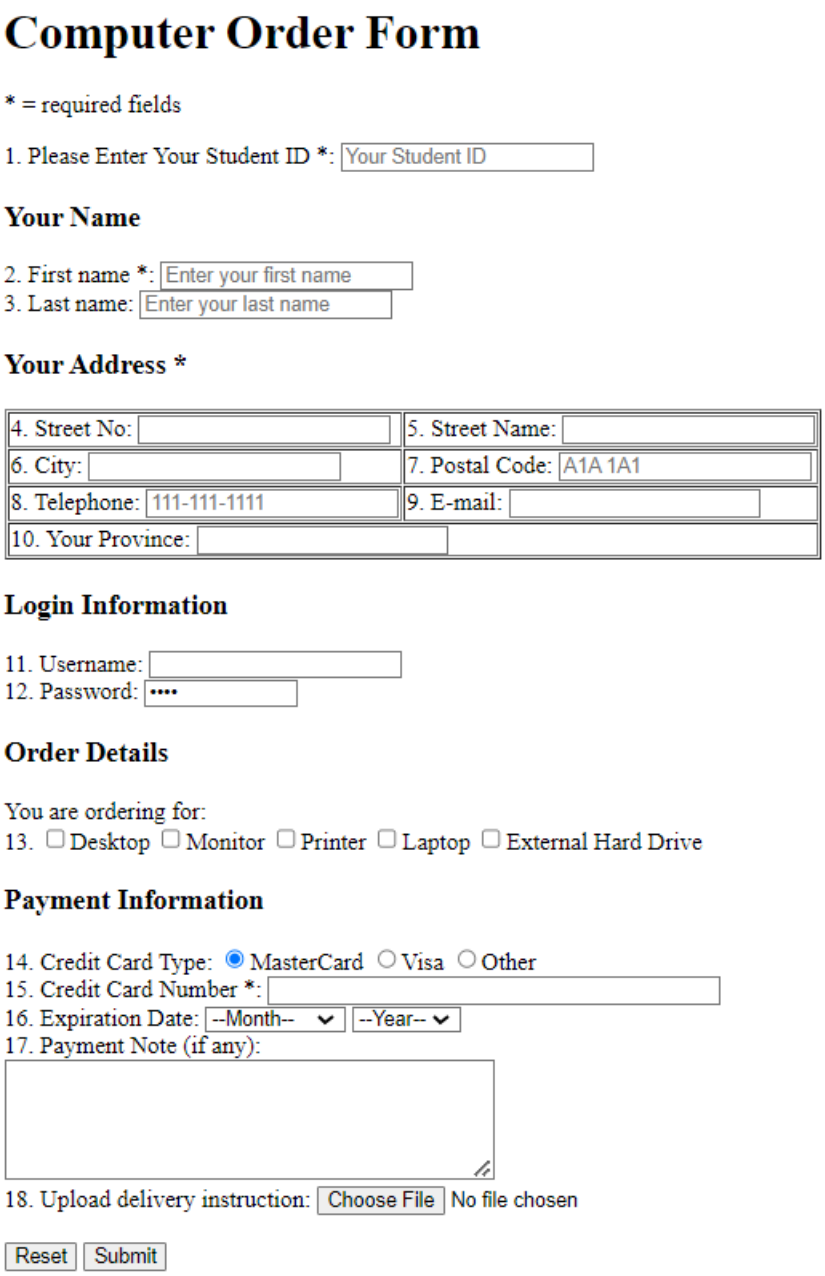
Please submit the description/context of a paragraph about your website, the domain, business logics/domain models and the scope of your website, ie: the bounded context.

The name of the file should be <Your First Name>_<FormName>.html.

The features should be equivalent with the computer order list below…

Develop the form:

You can choose and makeup suitable form fields and types that better suit with the type of data that will be collected through your form. See Figure 1 for an example form for 'Computer Order Form'. Use VS Code IDE as the editor for developing the form.



Computer Order Form

* = required fields

1. Please Enter Your Student ID *: Your Student ID

Your Name

2. First name *: Enter your first name
3. Last name: Enter your last name

Your Address *

4. Street No:	5. Street Name:
6. City:	7. Postal Code: A1A 1A1
8. Telephone: 111-111-1111	9. E-mail:
10. Your Province:	

Login Information

11. Username:
12. Password: ••••

Order Details

You are ordering for:
13. Desktop Monitor Printer Laptop External Hard Drive

Payment Information

14. Credit Card Type: MasterCard Visa Other
15. Credit Card Number *:
16. Expiration Date: --Month-- --Year--
17. Payment Note (if any):

18. Upload delivery instruction: Choose File No file chosen

Reset Submit

Figure 1: Computer Order Form

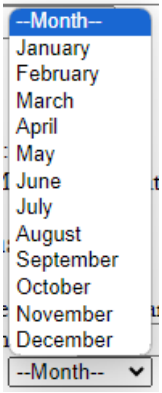
You **MUST** satisfy a set of requirements in your form as mentioned below and demonstrated in Figure 1.

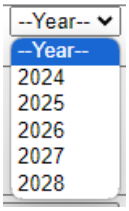
Important Requirements:

1. The form must have an appropriate title (e.g., 'Computer Order Form' in Figure 1).
2. You must have minimum 6 **required** fields in your form. All required fields must show an '*' with the label of the field. For example, Student ID and First Name fields in the Computer

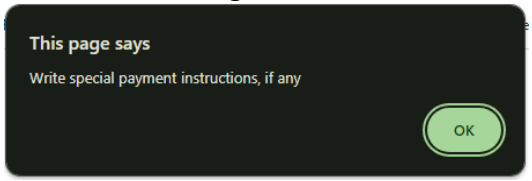
Order Form are required fields. You should demonstrate the use of required field validations for these fields. You can use HTML *required* attribute or JavaScript code.

3. The first field in the form should be set to **autofocus** (e.g., the autofocus of the form in Figure 1 is set to the first field (i.e., 1. Student ID field)).
4. There should be **placeholders** for at least 3 fields in the form (e.g., placeholder is added to 'Student ID', 'First name' and 'Last name' fields in the form in Figure 1)
5. There should be at least 10 fields receiving **text** input (e.g., 'Student ID', 'First name', 'Last name' and all address fields of the form in Figure 1 receive text inputs.)
6. At least two fields receive input following respective specific **patterns**. For example, Postal Code pattern = 'A1A 1A1' and Telephone pattern = '111-111-1111' is used in the form in Figure 1 (other possible patterns could be email, web url, etc.).
7. There should at least on field of **password** type receiving password as input (i.e., hiding the text as user enters in it). The password need to be validated with at least three characters, three digits, 2 special symbols and minimum length of 8. For example, 'Password' field in Figure 1 receives password type value.
8. There should at least one group of **checkbox** inputs allowing multiple option to be selected. For example, the Order Details inputs are checkboxes in Figure 1.
9. There should be at least one group of **radio button** inputs allowing options to be mutually selected (one of the options should be selected by default). For example, Credit Card Type inputs are mutually selected radio buttons with 'MasterCard' is selected by default.
10. There should be at least 2 different **select** fields – each containing at least 5 options. For example, 'Month' and 'Year' in Expiration Date field are two select fields in Figure 1 with option values as shown below.

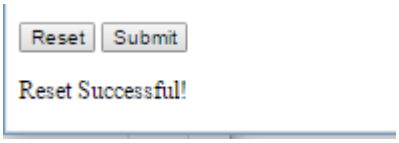




11. At least 1 field of type **textarea**, sized at least 5 rows and 40 columns. Display a suitable informative message in a pop-up window, once this field is **focused**. For example, the field 'Payment Note (if any)' in Figure 1 is a textarea accepting large text. It also displays "*Write special payment instruction, if any*" in a pop-up window once this field is focused as shown below when the form is rendered on Google Chrome browser.



12. There should be at least one **file upload** field allowing user to upload a file (e.g., the input # 18 in the Figure 1).
13. At least one set of related input fields should be arranged in a **table**. The table must have border. Demonstrate **row or column merging** in the table. For example, all fields receiving address information are placed in a table in Figure 1 and the last row spans multiple columns.
14. There should be one **Reset** button in the form allowing the user to reset all input. Write inline JavaScript code to



display the message "*Reset Successful*" right underneath the button, once the user clicks on this button, as follows:

15. The **Submit** button must be included in the form allowing the user to submit the form data (you do not need to provide any 'action' attribute for the form, since the backend script to receive form data is not in place). Write **in-document JavaScript** code to display the message "*Do you want to submit?*" in a separate pop-up window, once the user clicks on this button, as follows (when loaded on Google Chrome).

16. Add your Name and Sheridan student number as **commented** at the top in your HTML file.
17. Add suitable CSS effect on your form. Demonstrate the use of inline, in-document (i.e., internal) and external CSS. Choose your own unique style for your form.
18. Adjust the size of all fields as required (e.g., as shown in the example form).

Screenshots:

[**Screenshot**] Load your HTML form on Google Chrome browser. Take a screenshot of the page and paste it in the submission file.

[**Screenshots for the requirements**] You must provide screenshot(s) demonstrating all the above requirements. You **MUST** add appropriate requirement number(s) as label for each screenshot that it is satisfying. (You can use one screenshot showing multiple requirements – but don't forget to label it for the requirements it satisfies.) To demonstrate the requirements, you may need to take screenshots of the form rendered on browser or source code on VS Code.

Screenshot guideline must be followed to avoid potential penalties on earned grade.

Submission Guideline:

- Create a document file. The cover page must include following information:
  - Assignment No.: 1
  - Your full name
  - Your student ID:
  - Course Code: PROG43431
  - Instructor's name: Syed Tanbeer
- Paste screenshots for Part - I.
- Paste screenshots for Part – II.

Submit the document file with screenshots, the HTML file, and any external (JavaScript and CSS) file all as a single zipped file on the Dropbox for Assignment 1 on Slate. **Inclass demo for this assignment will be randomly selected during the lab time**..

---: End of Assignment 1:---