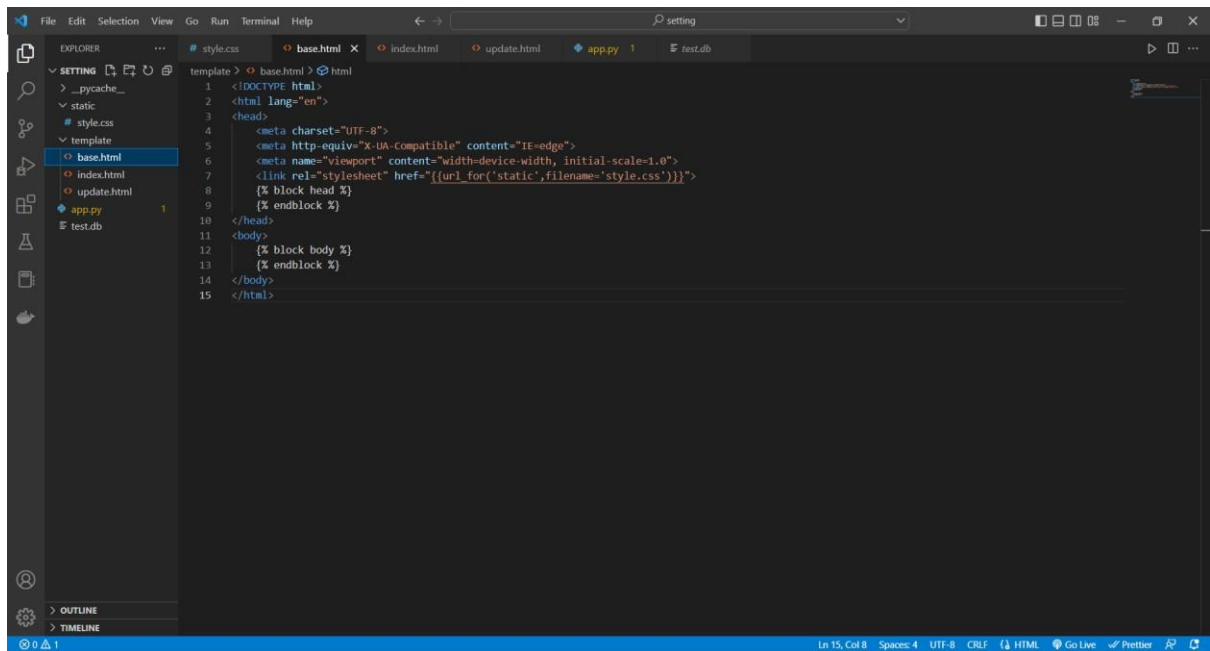


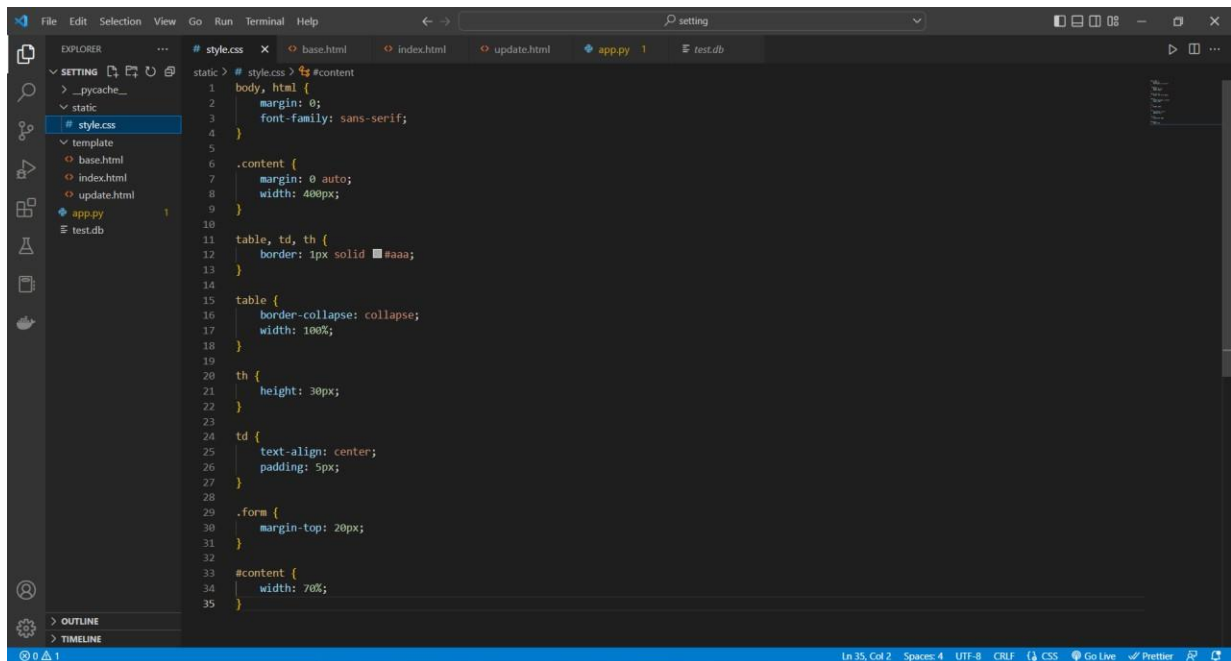
CREATE FLASK PROJECT

Date	18 November 2022
Team ID	PNT2022TMID24120
Project Name	Project – Plasma Donor Application



This screenshot shows the Visual Studio Code editor with a Flask project open. The Explorer sidebar on the left displays the project structure, including files like `__pycache__`, `static`, `style.css`, `template`, `base.html`, `index.html`, `update.html`, `app.py`, and `test.db`. The `base.html` file is selected and its content is visible in the main editor. The code is an HTML template with a head section containing meta tags for charset, compatibility, and viewport, and a link to the `style.css` file. The body section contains placeholders for blocks.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
8   {% block head %}
9   {% endblock %}
10 </head>
11 <body>
12   {% block body %}
13   {% endblock %}
14 </body>
15 </html>
```



This screenshot shows the Visual Studio Code editor with the same Flask project. The `style.css` file is now selected in the Explorer sidebar, and its content is displayed in the main editor. The CSS includes a `#content` selector for the main body, a `.content` selector for the content area, a `table` selector for a table, a `th` selector for table headers, a `td` selector for table data, a `.form` selector for a form, and another `#content` selector for a specific content area.

```
1 body, html {
2   margin: 0;
3   font-family: sans-serif;
4 }
5
6 .content {
7   margin: 0 auto;
8   width: 480px;
9 }
10
11 table, td, th {
12   border: 1px solid #aaa;
13 }
14
15 table {
16   border-collapse: collapse;
17   width: 100%;
18 }
19
20 th {
21   height: 30px;
22 }
23
24 td {
25   text-align: center;
26   padding: 5px;
27 }
28
29 .form {
30   margin-top: 20px;
31 }
32
33 #content {
34   width: 70%;
35 }
```

```
File Edit Selection View Go Run Terminal Help
setting

EXPLORER
  _pycache_
  static
  # style.css
  template
  base.html
  index.html
  update.html
  app.py
  test.db

template > index.html > ...
1 (% extends 'base.html' %)
2
3 {% block head %}
4 <title>Todo List</title>
5 {% endblock %}
6
7 {% block body %}
8 <div class="content">
9   <h1 style="text-align:center">Todo List</h1>
10   {% if tasks|length < 1% %}
11   <h1 style="text-align: center;">No task yet</h1>
12   {% else %}
13
14   <table>
15     <tr>
16       <th>Task</th>
17       <th>Date</th>
18       <th>Actions</th>
19     </tr>
20     {% for task in tasks %}
21
22     <tr>
23       <td>{{ task.content }}</td>
24       <td>{{ task.date_created.date() }}</td>
25       <td><a href="/delete/{{ task.id }}">Delete</a><br>
26         <a href="/update/{{ task.id }}">Update</a></td>
27     </tr>
28     {% endfor %}
29   </table>
30   {% endif %}
31 <br>
32
33 <form action="/" method="POST">
34   <input type="text" name="content" id="content">
35   <input type="submit" value="Add task">
36 </form>
37
38 </div>
39
40 {% endblock %}
```

```
File Edit Selection View Go Run Terminal Help
setting

EXPLORER
  _pycache_
  static
  # style.css
  template
  base.html
  index.html
  update.html
  app.py
  test.db

template > update.html > ...
1 (% extends 'base.html' %)
2
3 {% block head %}
4 <title>Task Master</title>
5 {% endblock %}
6
7 {% block body %}
8 <div class="content">
9   <h1 style="text-align: center;">Update Task</h1>
10
11   <div class="form">
12     <form action="/update/{{ task.id }}" method="POST">
13       <input type="text" name="content" id="content" value="{{ task.content }}">
14       <input type="submit" value="Update">
15     </form>
16   </div>
17 </div>
18
19 {% endblock %}
```

```
File Edit Selection View Go Run Terminal Help
# app.py
1 app = Flask(__name__)
2 app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///test.db'
3 db = SQLAlchemy(app)
4 db.init_app(app)
5
6 class Todo(db.Model):
7     id = db.Column(db.Integer, primary_key=True)
8     content = db.Column(db.String(100), nullable=False)
9     date_created = db.Column(db.DateTime, default=datetime.utcnow)
10
11     def __repr__(self):
12         return '<task %s>' % self.id
13
14 @app.route('/', methods = ['POST', 'GET'])
15 def index():
16     if request.method == 'POST':
17         #section "add"
18         task_content = request.form['content']
19         new_task = Todo(content=task_content)
20
21         #connecting to database
22         try:
23             db.session.add(new_task)
24             db.session.commit()
25             return redirect('/')
26         except:
27             return "there is an error"
28     else:
29         tasks = Todo.query.order_by(Todo.date_created).all()
30         return render_template("index.html", tasks = tasks)
31
32 #delete
33 @app.route('/delete/<int:id>')
34 def delete(id):
35     task_to_delete = Todo.query.get_or_404(id)
36
37     try:
38         db.session.delete(task_to_delete)
39         db.session.commit()
40         return redirect('/')
41     except:
42         return "there is an error in delete operation"
43
44 @update
```

4

127.0.0.1:5000



ToDo List
No task yet

ToDo List

		Actions
Complete frontend	2022-11-10	Delete Update
Complete backend	2022-11 10	oglp
<input type="text"/>		Add task

Update Task

[Update](#)

ToDo List

Task	Date	Actions
Write code	2D22-11-10	6' lq
<input type="text"/>		add a r.