

IS601 Final Project

Step1 Creating Your First Flask Application

In this step, create first flask template with basic code

13 lines (9 sloc) | 205 Bytes

```
1 from flask import Flask, render_template
2
3
4 # Create Flask's `app` object
5 app = Flask(
6     __name__,
7     template_folder="templates"
8 )
9
10
11 @app.route("/")
12 def hello():
13     return render_template("index.html")
```

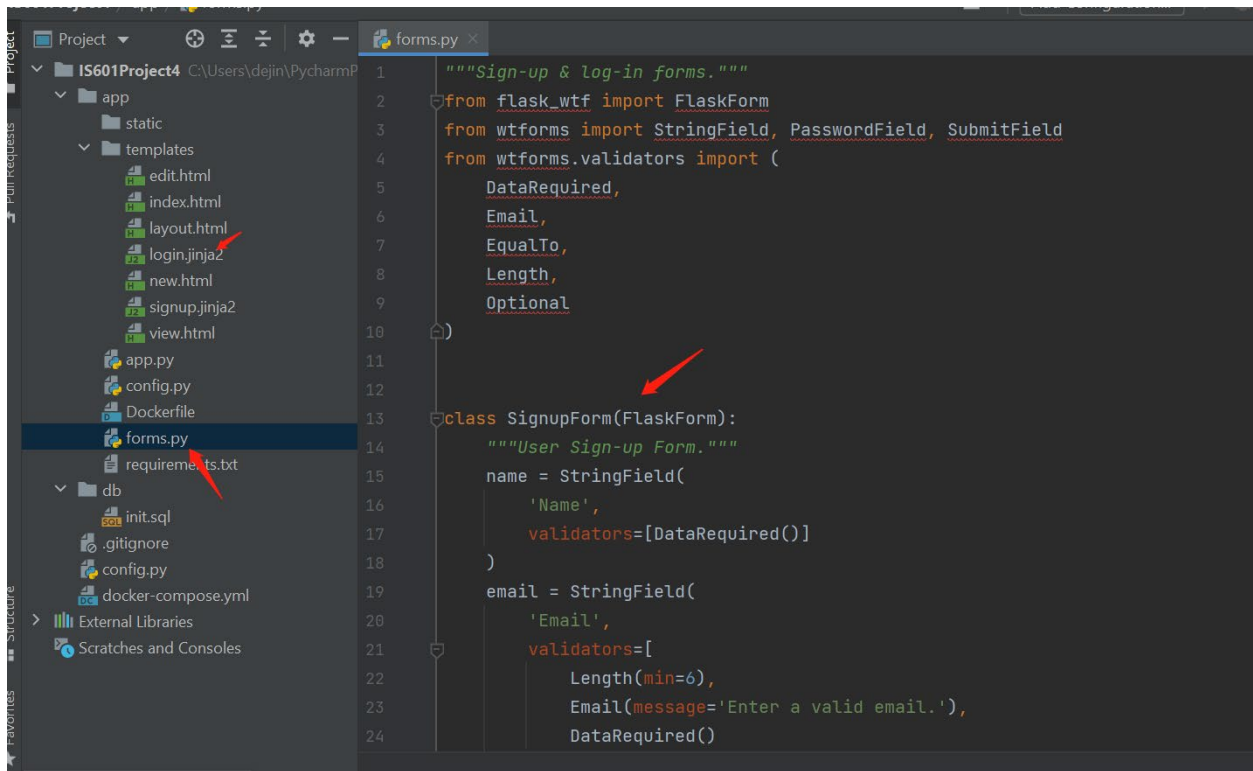
Step2 Rendering Pages in Flask Using Jinja

In this step, create front html with jinja

```
1 {% extends "layout.html" %}
2 {% block content %}
3 <table class="table">
4     <thead>
5     <tr>
6         <th scope="col">Index</th>
7         <th scope="col">Year</th>
8         <th scope="col">Age</th>
9         <th scope="col">Name</th>
10        <th scope="col">Movie</th>
11        <th scope="col">Note</th>
12    </tr>
13    </thead>
14    <tbody>
15    {% for osc in oscar %}
16    <tr>
17        <td><a href="view {{osc.id}}>{{ osc.sIndex }}</a></td>
18        <td>{{ osc.sYear }}</td>
19        <td>{{ osc.sAge }}</td>
20        <td>{{ osc.sName }}</td>
21        <td>{{ osc.sMovie }}</td>
22        <td>{{ osc.sNote }}</td>
23    </tr>
24    {% endfor %}
```

Step3. Handling Forms in Flask with Flask-WTF

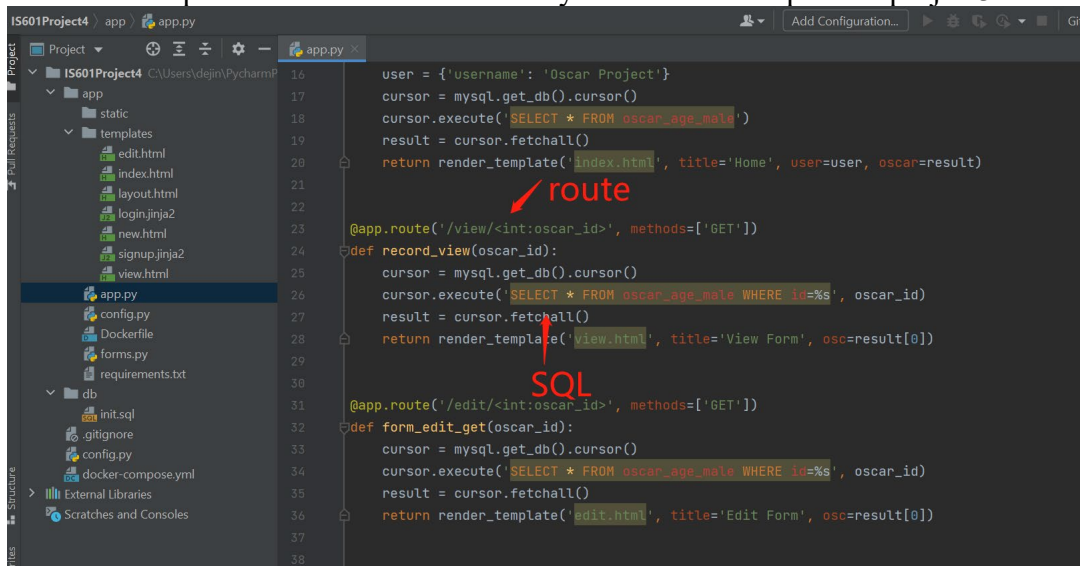
In this step, build login and sign up template forms and jinja for later use.



Step4 The Art of Routing in Flask

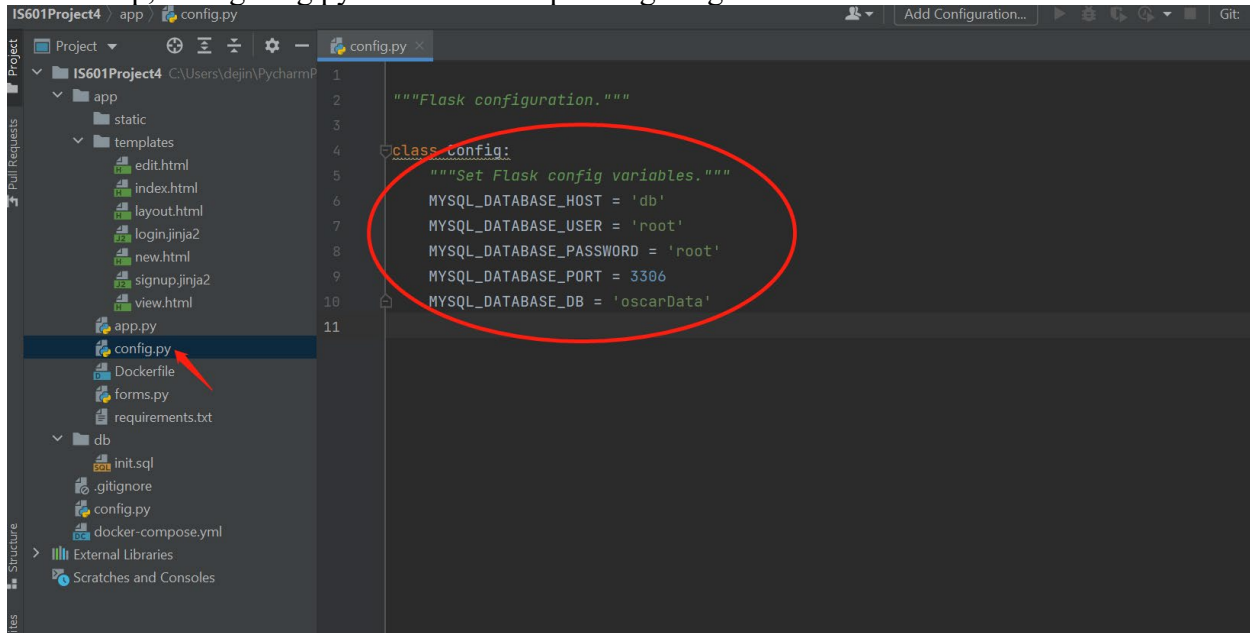
Step 9 Connect Flask to a Database with Flask-SQLAlchemy

These two steps are combined as we already build these two parts in project3



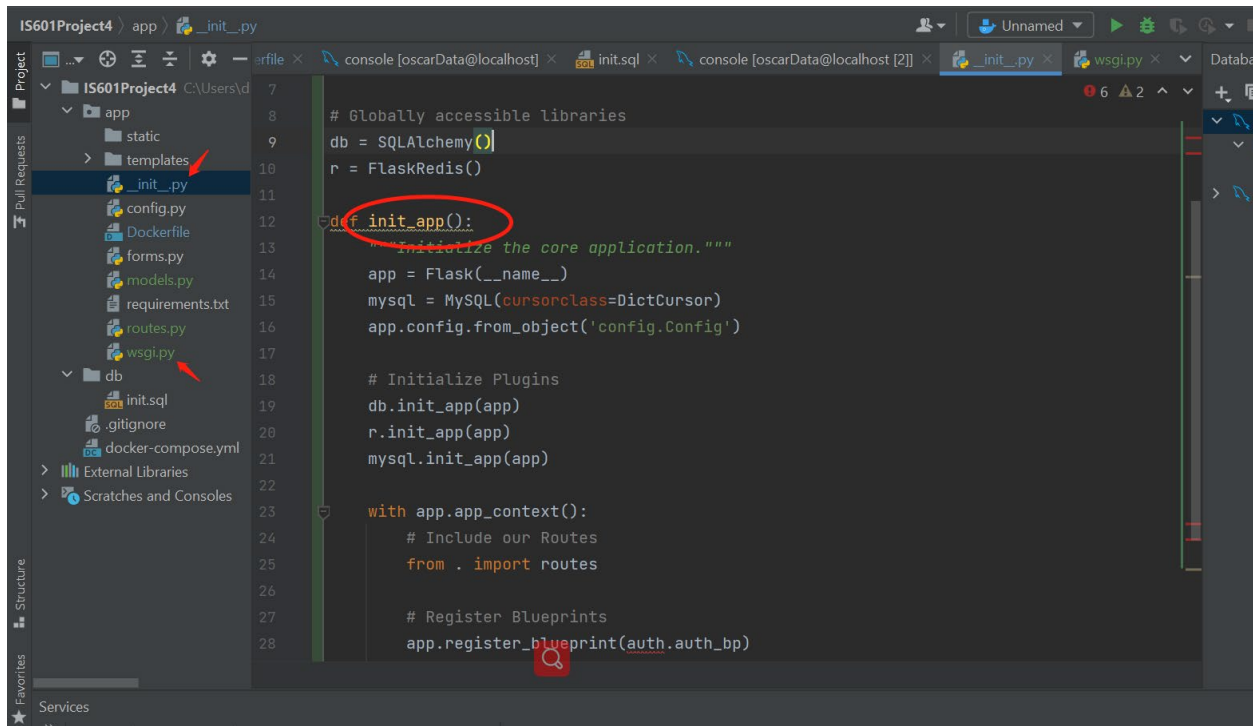
Step5. Configuring Your Flask App

In this step, configuring py are built and sql configuring coded.



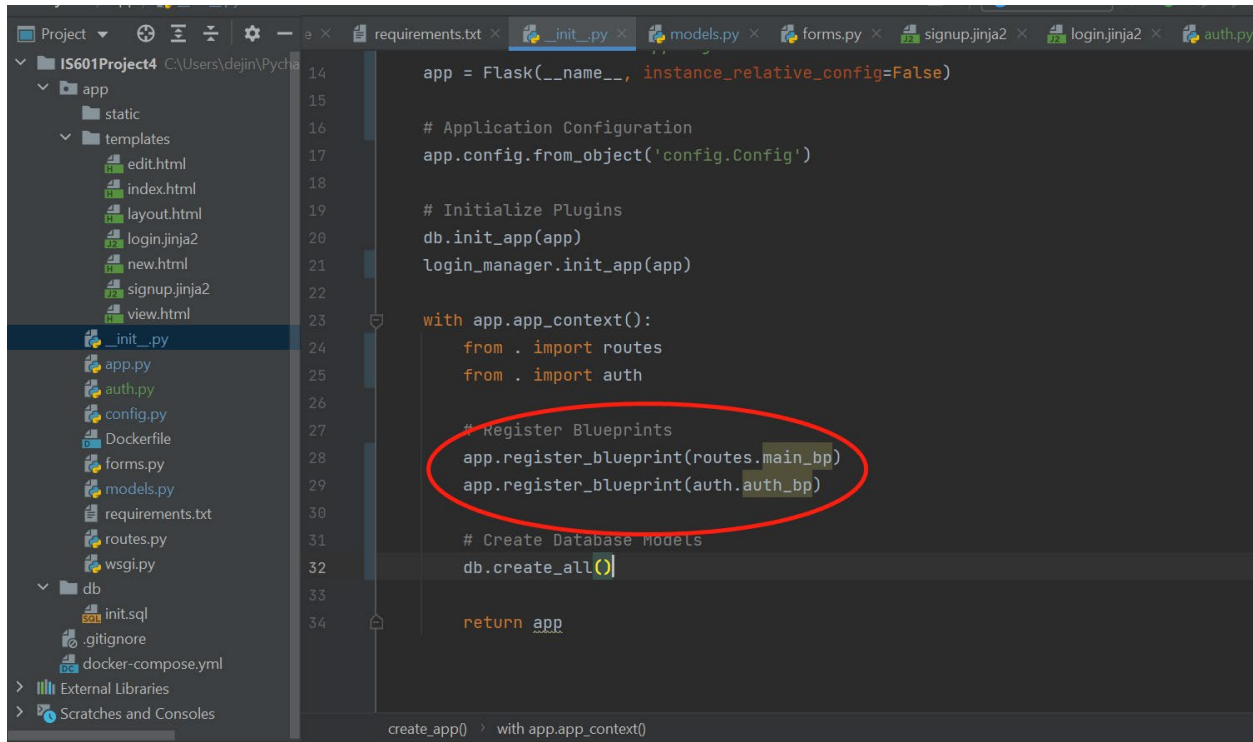
Step6. Demystifying Flask's Application Factory

In this step, app.py was separated to organized .py files, as __init__.py , wsgi.py, and routes.py



Step7. Organizing Flask Apps with Blueprints

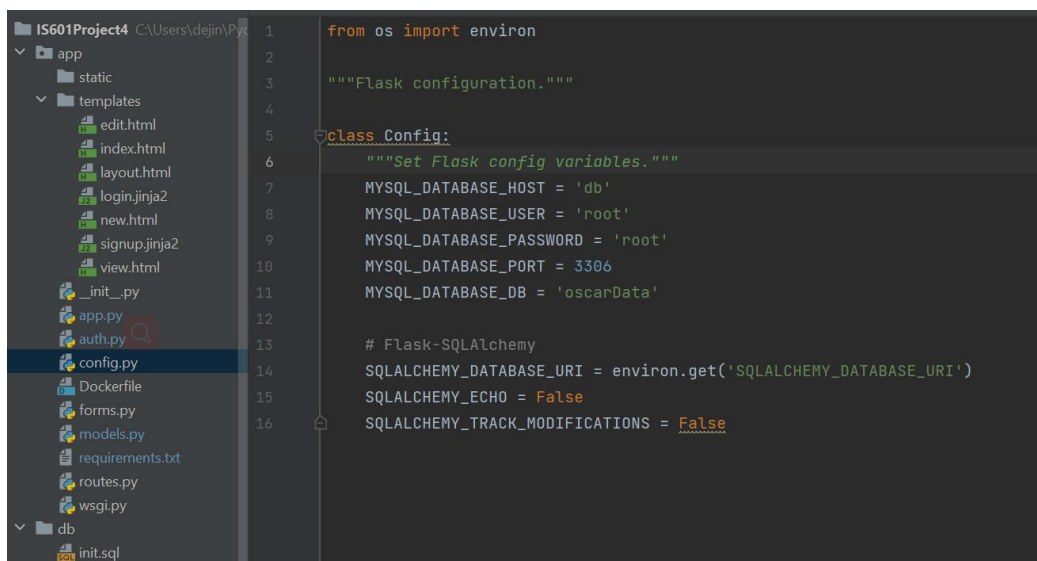
In this step, we use blueprint to manage app structure.



```
14 app = Flask(__name__, instance_relative_config=False)
15
16 # Application Configuration
17 app.config.from_object('config.Config')
18
19 # Initialize Plugins
20 db.init_app(app)
21 login_manager.init_app(app)
22
23 with app.app_context():
24     from . import routes
25     from . import auth
26
27     # Register Blueprints
28     app.register_blueprint(routes.main_bp)
29     app.register_blueprint(auth.auth_bp)
30
31 # Create Database Models
32 db.create_all()
33
34 return app
```

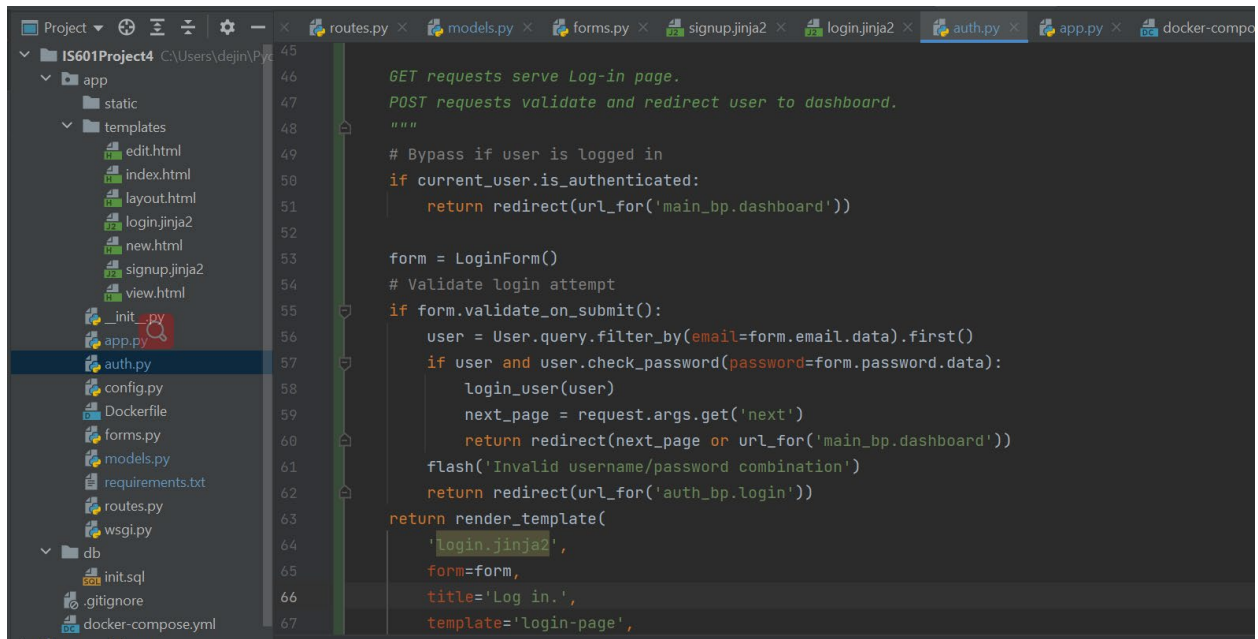
Step10. Handle User Accounts & Authentication in Flask with Flask-Login

Config screenshot



```
1 from os import environ
2
3 """Flask configuration."""
4
5 class Config:
6     """Set Flask config variables."""
7
8     MYSQL_DATABASE_HOST = 'db'
9     MYSQL_DATABASE_USER = 'root'
10    MYSQL_DATABASE_PASSWORD = 'root'
11    MYSQL_DATABASE_PORT = 3306
12    MYSQL_DATABASE_DB = 'oscarData'
13
14    # Flask-SQLAlchemy
15    SQLALCHEMY_DATABASE_URI = environ.get('SQLALCHEMY_DATABASE_URI')
16    SQLALCHEMY_ECHO = False
17    SQLALCHEMY_TRACK_MODIFICATIONS = False
```

Auth.py for login confirmation



```
45 GET requests serve Log-in page.
46 POST requests validate and redirect user to dashboard.
47
48
49 # Bypass if user is logged in
50 if current_user.is_authenticated:
51     return redirect(url_for('main_bp.dashboard'))
52
53 form = LoginForm()
54 # Validate login attempt
55 if form.validate_on_submit():
56     user = User.query.filter_by(email=form.email.data).first()
57     if user and user.check_password(password=form.password.data):
58         login_user(user)
59         next_page = request.args.get('next')
60         return redirect(next_page or url_for('main_bp.dashboard'))
61     flash('Invalid username/password combination')
62     return redirect(url_for('auth_bp.login'))
63 return render_template(
64     'login.jinja2',
65     form=form,
66     title='Log in.',
67     template='login-page',
```

Step11. Managing Session Data with Flask-Session & Redis

In this step, session was added into application, screenshot shows session in routes.py


```
156 '/login.jinja2',
157 title='Create an Account.',
158 form=LoginForm,
159 template='login-page',
160 body="Log in to your account."
161 )
162
163 @app.route("/session", methods=["GET"])
164 @login_required
165 def session_view():
166     """Display session variable value."""
167     return render_template(
168         "session.jinja2",
169         title="MLB Players",
170         template="dashboard-template",
171         session_variable=str(session["redis_test"]),
172     )
173
174
175
176 @app.route("/logout")
177 @login_required
178 def logout():
179     logout0
```