

APPLICATION BUILDING – BUILDING PYTHON CODE

Date	16 November 2022
Team ID	PNT2022TMID29238
Project Name	Project - A Gesture-based Tool for Sterile Browsing of Radiology Images

```
1  import os
2  from flask import Flask, render_template, url_for, redirect
3  from flask_sqlalchemy import SQLAlchemy #DB
4  from flask_login import UserMixin, login_user, login_manager, login_required, logout_user, current_user
5  from flask_wtf import FlaskForm
6  from wtforms import StringField, PasswordField, SubmitField, IntegerField
7  from wtforms.validators import InputRequired, Length, ValidationError
8  from flask_bcrypt import Bcrypt
9
10 app = Flask(__name__)
11
12
13 db = SQLAlchemy(app)
14 bcrypt = Bcrypt(app)
15 app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///database.db'
16 app.config['SECRET_KEY'] = 'thisisasecretkey'
17
18
19 login_manager = LoginManager()
20 login_manager.init_app(app)
21 login_manager.login_view = 'login'
22
23
24 @login_manager.user_loader
25 def load_user(user_id):
26     return User.query.get(int(user_id))
27
28
29 class User(db.Model, UserMixin):
30     id = db.Column(db.Integer, primary_key=True)
31     username = db.Column(db.String(20), nullable=False, unique=True)
32     password = db.Column(db.String(80), nullable=False)
33     #phonenumber = db.Column(db.String(10), nullable=False, unique=True)
34
35
36 class RegisterForm(FlaskForm):
```

```
38     InputRequired(), length(min=4, max=20)), render_kw={"placeholder": "Enter your username"})
39
40 password = PasswordField("Password : ", validators=[
41     InputRequired(), length(min=8, max=20)], render_kw={"placeholder": "Enter your password"})
42
43
44
45 submit = SubmitField("Signup")
46
47 def validate_username(self, username):
48     existing_user_username = User.query.filter_by(username=username.data).first()
49     if existing_user_username:
50         raise ValidationError('That username already exists. Please choose a different one.')
51
52
53 class LoginForm(FlaskForm):
54     username = StringField(validators=[
55         InputRequired(), length(min=4, max=20)], render_kw={"placeholder": "Username"})
56
57 password = PasswordField(validators=[
58     InputRequired(), length(min=8, max=20)], render_kw={"placeholder": "Password"})
59
60 submit = SubmitField("Login")
61
62
63 @app.route('/')
64 def home():
65     return render_template('home.html')
66
67 @app.route('/about')
68 def about():
69     return render_template('about.html')
70
71 @app.route('/login', methods=['GET', 'POST'])
72 def login():
73     form = LoginForm()
74     if form.validate_on_submit():
```

```

75     user = User.query.filter_by(username=form.username.data).first()
76     if user:
77         if bcrypt.check_password_hash(user.password, form.password.data):
78             login_user(user)
79             return redirect(url_for('dashboard'))
80     return render_template('login.html', form=form)
81
82
83 @app.route('/dashboard', methods=['GET', 'POST'])
84 @login_required
85 def dashboard():
86     return render_template('dashboard.html')
87
88
89 @app.route('/logout', methods=['GET', 'POST'])
90 @login_required
91 def logout():
92     logout_user()
93     return redirect(url_for('home'))
94
95
96 @app.route('/register', methods=['GET', 'POST'])
97 def register():
98     form = RegisterForm()
99
100     if form.validate_on_submit():
101         hashed_password = bcrypt.generate_password_hash(form.password.data)
102         new_user = User(username=form.username.data, password=hashed_password)
103         db.session.add(new_user)
104         db.session.commit()
105         return redirect(url_for('login'))
106
107     return render_template('register.html', form=form)
108
109
110 if __name__ == '__main__':
111     app.run(debug=True, port = 5000)

```