

```
1 from flask import Flask, render_template, redirect,
   url_for, flash, request
2 from flask_wtf import FlaskForm
3 from wtforms import StringField, PasswordField,
   SubmitField, FloatField
4 from wtforms.validators import DataRequired, Length,
   Email, EqualTo
5 from flask_login import LoginManager, UserMixin,
   login_user, login_required, logout_user, current_user
6 from werkzeug.security import generate_password_hash
   , check_password_hash
7 import os
8
9 # Flask app setup
10 app = Flask(__name__, template_folder="templates")
11 app.secret_key = os.environ.get('SECRET_KEY') or os.
   urandom(24)
12
13 # In-memory user storage (for demonstration only -
   use a real database in production)
14 users = {}
15
16 # Flask-Login setup
17 login_manager = LoginManager(app)
18 login_manager.login_view = 'login'
19
20 # User class for Flask-Login
21 class User(UserMixin):
22     def __init__(self, id):
23         self.id = id
24
25 @login_manager.user_loader
26 def load_user(user_id):
27     return User(user_id) if user_id in users else
   None
28
29 # Flask-WTF Forms
30 class RegisterForm(FlaskForm):
31     username = StringField('Username', validators=[
   DataRequired(), Length(min=2, max=20)])
32     email = StringField('Email', validators=[
```

```

32 DataRequired(), Email()])
33     password = PasswordField('Password', validators=[
    DataRequired()])
34     confirm_password = PasswordField('Confirm
    Password', validators=[DataRequired(), EqualTo('
    password')])
35     submit = SubmitField('Register')
36
37 class LoginForm(FlaskForm):
38     username = StringField('Username', validators=[
    DataRequired()])
39     password = PasswordField('Password', validators=[
    DataRequired()])
40     submit = SubmitField('Login')
41
42 class PredictForm(FlaskForm):
43     input_data = FloatField('Enter Data (e.g., price
    prediction input)', validators=[DataRequired()])
44     submit = SubmitField('Predict Price')
45
46 # Routes
47 @app.route('/')
48 def home():
49     return render_template('index.html')
50
51 @app.route('/register', methods=['GET', 'POST'])
52 def register():
53     form = RegisterForm()
54     if form.validate_on_submit():
55         username = form.username.data
56         email = form.email.data
57         password_hash = generate_password_hash(form.
    password.data)
58
59         if username in users:
60             flash('Username already exists. Please
    choose a different one.', 'danger')
61         elif any(user['email'] == email for user in
    users.values()):
62             flash('Email already used. Please use a
    different one.', 'danger')

```

```

63         else:
64             users[username] = {'email': email, '
password': password_hash}
65             flash('Registration successful! You can
now log in.', 'success')
66             return redirect(url_for('login'))
67         return render_template('register.html', form=
form)
68
69 @app.route('/login', methods=['GET', 'POST'])
70 def login():
71     form = LoginForm()
72     if form.validate_on_submit():
73         username = form.username.data
74         password = form.password.data
75
76         user = users.get(username)
77         if user and check_password_hash(user['
password'], password):
78             user_obj = User(username)
79             login_user(user_obj)
80             flash('Login successful!', 'success')
81             return redirect(url_for('dashboard'))
82         else:
83             flash('Login failed. Check your username
and password.', 'danger')
84         return render_template('login.html', form=form)
85
86 @app.route('/dashboard')
87 @login_required
88 def dashboard():
89     return render_template('dashboard.html',
username=current_user.id)
90
91 @app.route('/predict', methods=['GET', 'POST'])
92 def predict():
93     form = PredictForm()
94     prediction_result = None
95     if form.validate_on_submit():
96         input_data = form.input_data.data
97         # Simple prediction logic (e.g., price per

```

```
97 square foot)
98         price_per_square_foot = 200 # Example value
99         prediction_result = input_data *
        price_per_square_foot
100         flash('Prediction successful!', 'success')
101         return render_template('predict.html', form=form
        , result=prediction_result)
102
103 @app.route('/logout')
104 @login_required
105 def logout():
106     logout_user()
107     flash('You have been logged out.', 'info')
108     return redirect(url_for('home'))
109
110 # Run the app
111 if __name__ == '__main__':
112     app.run(debug=True)
```