

1

```

config.py x
1  import os
2
3  basedir = os.path.abspath(os.path.dirname(__file__))
4
5  WTF_CSRF_ENABLED = True
6  SECRET_KEY = 'safdfsdf'
7
8
9  SQLALCHEMY_DATABASE_URI = 'sqlite:/// ' + os.path.join(basedir, 'site.db')
10 SQLALCHEMY_TRACK_MODIFICATIONS = False

```

3

```

16  message = db.Column(db.Text, nullable=False)
17  created_at = db.Column(db.DateTime, default=datetime.utcnow)
18
19  def repr(self):
20      return f"Feedback('{self.name}', '{self.message}', '{self.created_at}')"
21
22  class User(db.Model):
23      id = db.Column(db.Integer, primary_key=True)
24      username = db.Column(db.String(20), unique=True, nullable=False)
25      email = db.Column(db.String(120), unique=True, nullable=False)
26      image_file = db.Column(db.String(20), nullable=False, default='default.jpg')
27      password = db.Column(db.String(60), nullable=False)
28
29      def repr(self):
30          return f"User('{self.username}', '{self.email}')"

```

4

DB Browser for SQLite - D:\labss\lab7\app\site.db

Файл Редагування Вид Tools Довідка

Нова база даних Відкрити базу даних Записати зміни Скасувати зміни Open Project Зберегти проект Attach Database Закрити базу даних

Структура БД Переглянути дані Редагувати прагми Виконати SQL

Таблиця: user Filter in any column

	id	username	email	image_file	password
	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
1	1	user1	pavliuk@example....	default.jpg	password1
2	2	user2	yulia2@example.c...	default.jpg	password2
3	3	user3	user3@example.com	default.jpg	password3

DB Browser for SQLite - D:\labss\lab7\app\site.db

Файл Редагування Вид Tools Довідка

Нова база даних Відкрити базу даних Залучити зміни Скасувати зміни Open Project Зберегти проект Attach Database Закрити базу даних

Структура БД Переглянути дані Редагувати прапну Виконати SQL

Таблиця: user

	id	username	email	image_file	password
Філт...	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
1	1	user1	pavliuk@example....	default.jpg	12345678
2	2	user2	yulia2@example.c...	default.jpg	password2

5

```

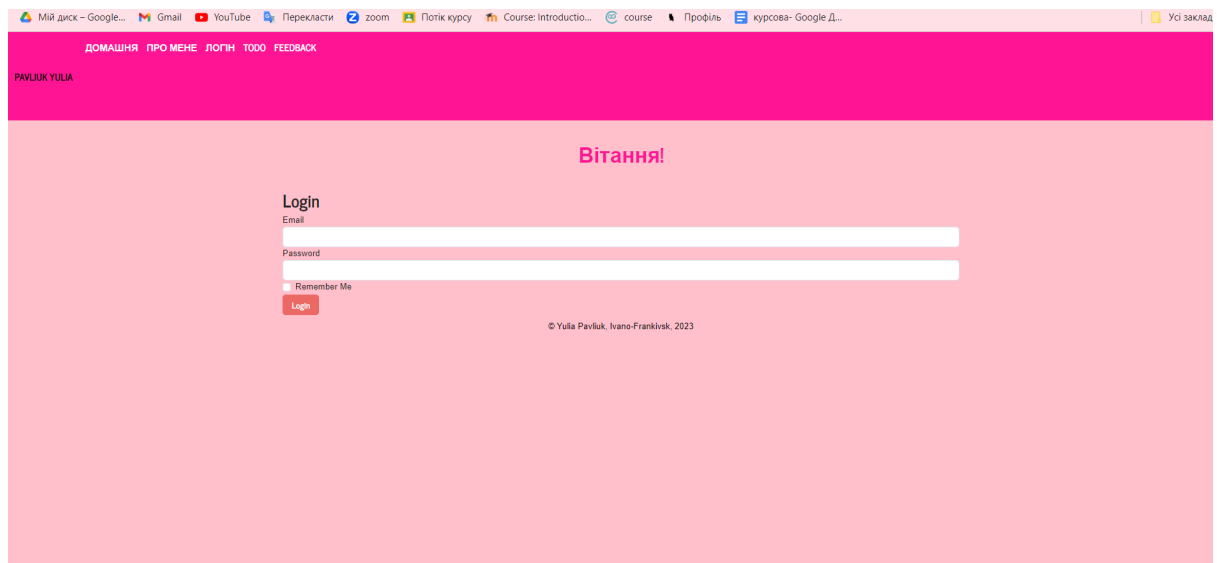
1 from flask_wtf import FlaskForm
2 from wtforms import StringField, PasswordField, BooleanField, SubmitField, TextAreaField
3 from wtforms.validators import InputRequired, Length, DataRequired, Email, EqualTo
4
5 2 usages
6 class FeedbackForm(FlaskForm):
7     name = StringField('Your Name', validators=[InputRequired(), Length(max=50)], render_kw={"class": "form-control"})
8     message = TextAreaField('Your Feedback', validators=[InputRequired()], render_kw={"class": "form-control"})
9     submit = SubmitField('Submit Feedback', render_kw={"class": "btn btn-primary"})
10
11 class RegistrationForm(FlaskForm):
12     username = StringField('Username', validators=[DataRequired(), Length(min=4, max=14)])
13     email = StringField('Email', validators=[DataRequired(), Email()])
14     password = PasswordField('Password', validators=[DataRequired(), Length(min=6)])
15     confirm_password = PasswordField('Confirm Password', validators=[DataRequired(), EqualTo('password')])
16     submit = SubmitField('Sign up')
17
18 3 usages
19 class LoginForm(FlaskForm):
20     email = StringField('Email', validators=[DataRequired(), Email()])
21     password = PasswordField('Password', validators=[DataRequired()])
22     remember = BooleanField('Remember Me')
23     submit = SubmitField('Login')

```

6

```
13 from app.models import Todo, Feedback
14
15 auth_data_path = join(dirname(abspath(__file__)), 'users.json')
16 with open(auth_data_path, 'r') as f:
17     auth_data = json.load(f)
18
19 @app.route("/register", methods=['GET', 'POST'])
20 def register():
21     form = RegistrationForm()
22     if form.validate_on_submit():
23         flash(message=f'Account created for {form.username.data}!', category='success')
24         return redirect(url_for('login'))
25     return render_template(template_name_or_list='register.html', form=form, title='Register')
26
27 @app.route("/login", methods=['GET', 'POST'])
28 def login():
29     form = LoginForm()
30     if form.validate_on_submit():
31         if form.email.data == 'test@gmail.com' and form.password.data == '111111':
32             flash(message='You have been logged in!', category='success')
33             return redirect(url_for('index')) # Потрібно вказати правильний endpoint, можливо, 'index'
34         else:
35             flash(message='Login unsuccessful. Please check username and password', category='warning')
36     return render_template(template_name_or_list='login.html', form=form, title='Login')
37
38
39 @app.route('/feedback', methods=['GET', 'POST'])
```

7





9



Структура БД Переглянути дані Редагувати прагну Виконати SQL					
Таблиця: user					
	id	username	email	image_file	password
	Філ...	Фільтр	Фільтр	Фільтр	Фільтр
1	1	user1	pavliuk@example.com	default.jpg	12345678
2	2	user2	yulia2@example.com	default.jpg	password2
3	3	Yulia	yuliia.pavliuk.21@pnu.edu.ua	default.jpg	123456

10

```

22 class User(db.Model):
23     id = db.Column(db.Integer, primary_key=True)
24     username = db.Column(db.String(20), unique=True, nullable=False)
25     email = db.Column(db.String(120), unique=True, nullable=False)
26     image_file = db.Column(db.String(20), nullable=False, default='default.jpg')
27     password = db.Column(db.String(60), nullable=False)
28
29     1 usage (1 dynamic)
29     def set_password(self, password):
30         self.password = bcrypt.generate_password_hash(password).decode('utf-8')
31
32     def check_password(self, password):
33         return bcrypt.check_password_hash(self.password, password)
34
35     def repr(self):
36         return f"User('{self.username}', '{self.email}')"

```

```

16 with open(auth_data_path, 'r') as f:
17     auth_data = json.load(f)
18
19 @app.route("/register", methods=['GET', 'POST'])
20 def register():
21     form = RegistrationForm()
22     if form.validate_on_submit():
23         hashed_password = bcrypt.generate_password_hash(form.password.data).decode('utf-8')
24         user = User(username=form.username.data, email=form.email.data, password=hashed_password)
25         db.session.add(user)
26         db.session.commit()
27         flash(message='Your account has been created!', category='success')
28         return redirect(url_for('login'))
29     return render_template(template_name_or_list='register.html', title='Register', form=form)
30
31 @app.route("/login", methods=['GET', 'POST'])
32 def login():

```

Таблиця: user

	id	username	email	image_file	password
Фільтр...	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
1	1	user1	pavliuk@example.com	default.jpg	12345678
2	2	user2	yulia2@example.com	default.jpg	password2
3	3	Yulia	yulia.pavliuk21@pnu.edu.ua	default.jpg	123456
4	4	user	user@gmail.com	default.jpg	\$2b\$12\$...

Результат: Текст

1 \$2b\$12\$EbnWqR/vbny2d1yCVGJ.vM/1y3olaByMctMQK6Gf6whAI3aKR...

Тип даних у коді: Текст / Число
60 символів

Застосувати

```
submit = SubmitField('Sign up')

def validate_username(self, username):
    user = User.query.filter_by(username=username.data).first()
    if user:
        raise ValidationError('Це ім'я користувача вже зайняте. Виберіть інше.')

1 usage (1 dynamic)
def validate_email(self, email):
    user = User.query.filter_by(email=email.data).first()
    if user:
        raise ValidationError('Цей email вже зареєстрований. Використайте інший.')
```

```
1 usage (1 dynamic)
def validate_email(self, email):
    user = User.query.filter_by(email=email.data).first()
    if user:
        raise ValidationError('Цей email вже зареєстрований. Використайте інший.')
```

```
def validate_usernameformat(self, username):
    if not Regexp(r'^[a-zA-Z0-9]+$'):
        raise ValidationError('Ім'я користувача може містити лише літери, цифри та знак підкреслення.')(self, username)
    raise ValidationError('Ім'я користувача може містити лише літери, цифри та знак підкреслення.')
```

ДОМАШНЯ | ПРО МЕНЕ | ЛОГІН | REGISTER | TODO | FEEDBACK | USERS

PAVLIUK YULIA

Вітання!

All Users

user1

user2

Yulia

user

Total users: 4

© Yulia Pavliuk, Ivano-Frankivsk, 2023

Таблица: user					
	id	username	email	image_file	password
	Філ...	Фільтр	Фільтр	Фільтр	Фільтр
1	1	user1	pavliuk@example.com	default.jpg	12345678
2	2	user2	yulia2@example.com	default.jpg	password2
3	3	Yulia	yuliia.pavliuk.21@pnu.edu.ua	default.jpg	123456
4	4	user	user@gmail.com	default.jpg	\$2b\$12\$...