Міністерство освіти і науки України

Державний університет ,,Житомирська політехніка”

Кафедра ФІКТ

Група:ВТ-21-1

Програмування мовою Python

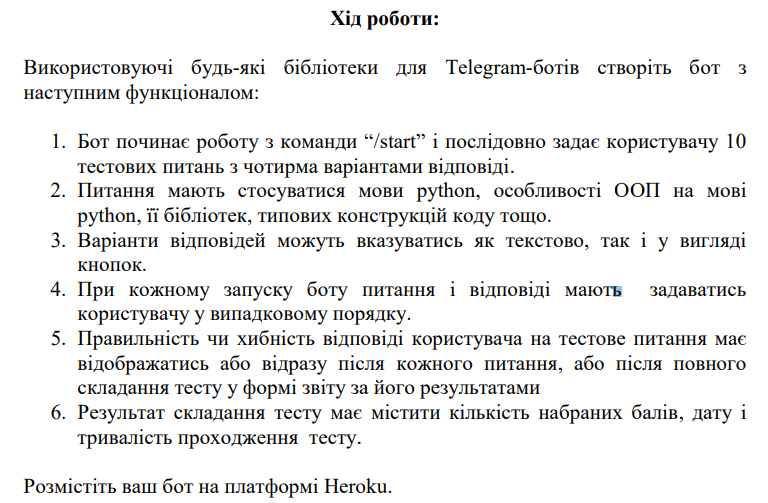
Лабораторна робота №12

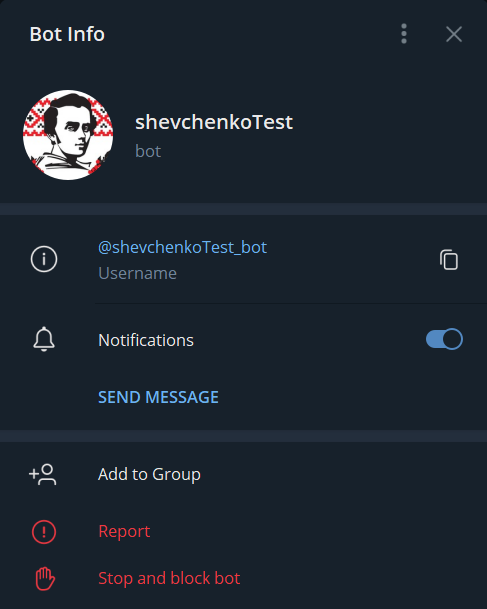
«Telegram-бот»

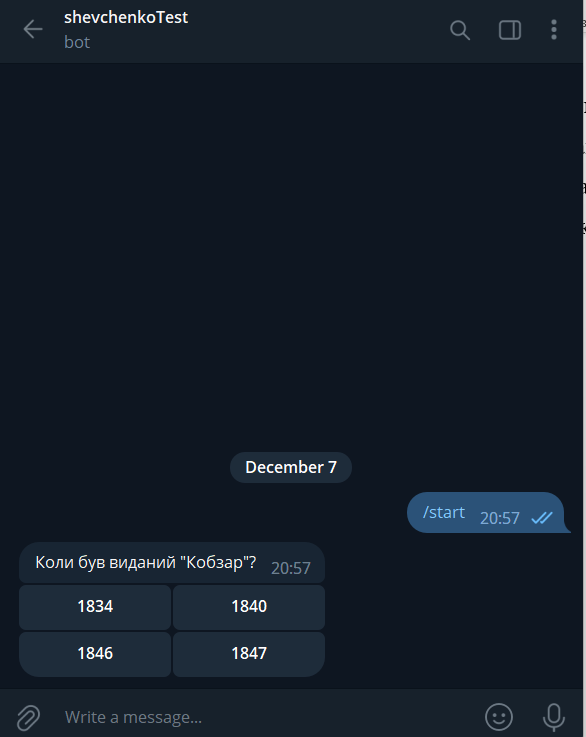
Виконав: Вигнич О. С.

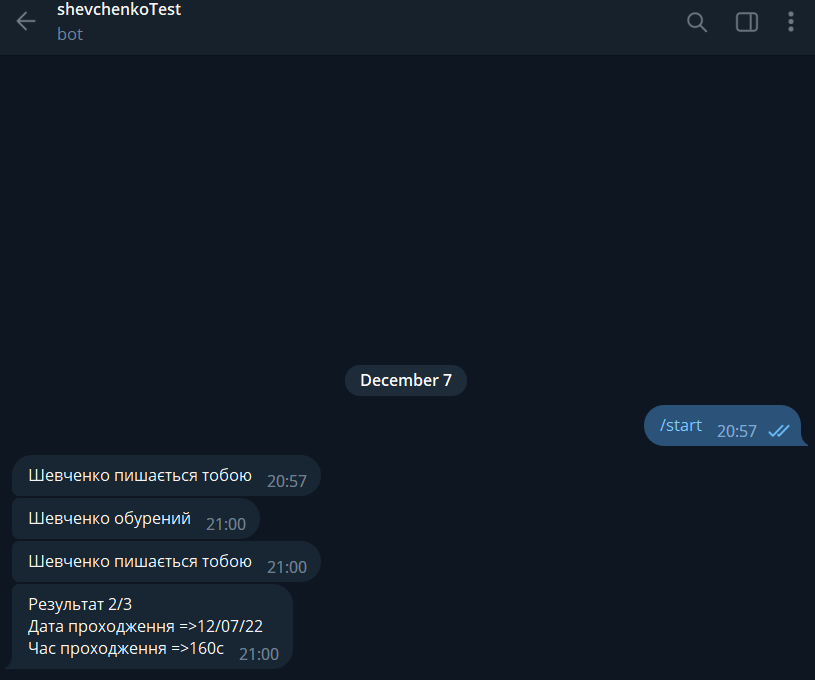
Прийняв: Морозов Д. С.

**Мета роботи:** ознайомитися з роботою з бібліотек для написання ботів для месенджера Telegram

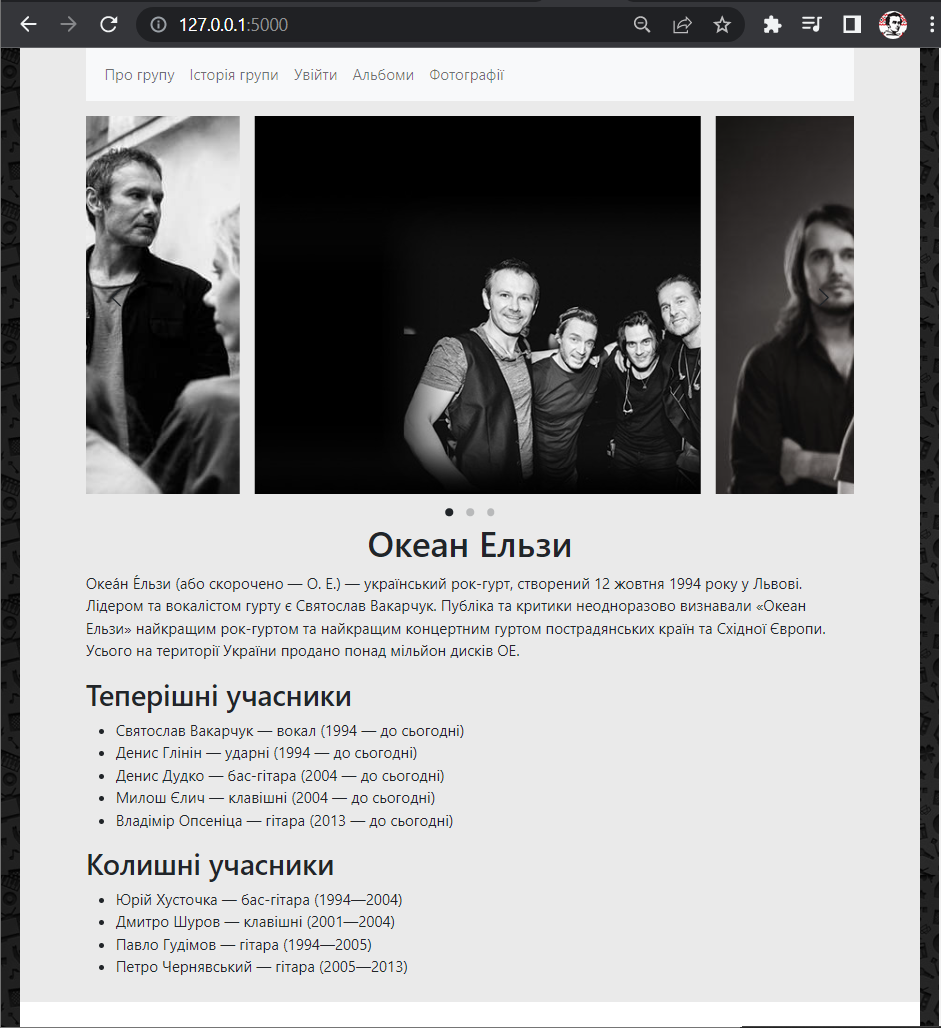


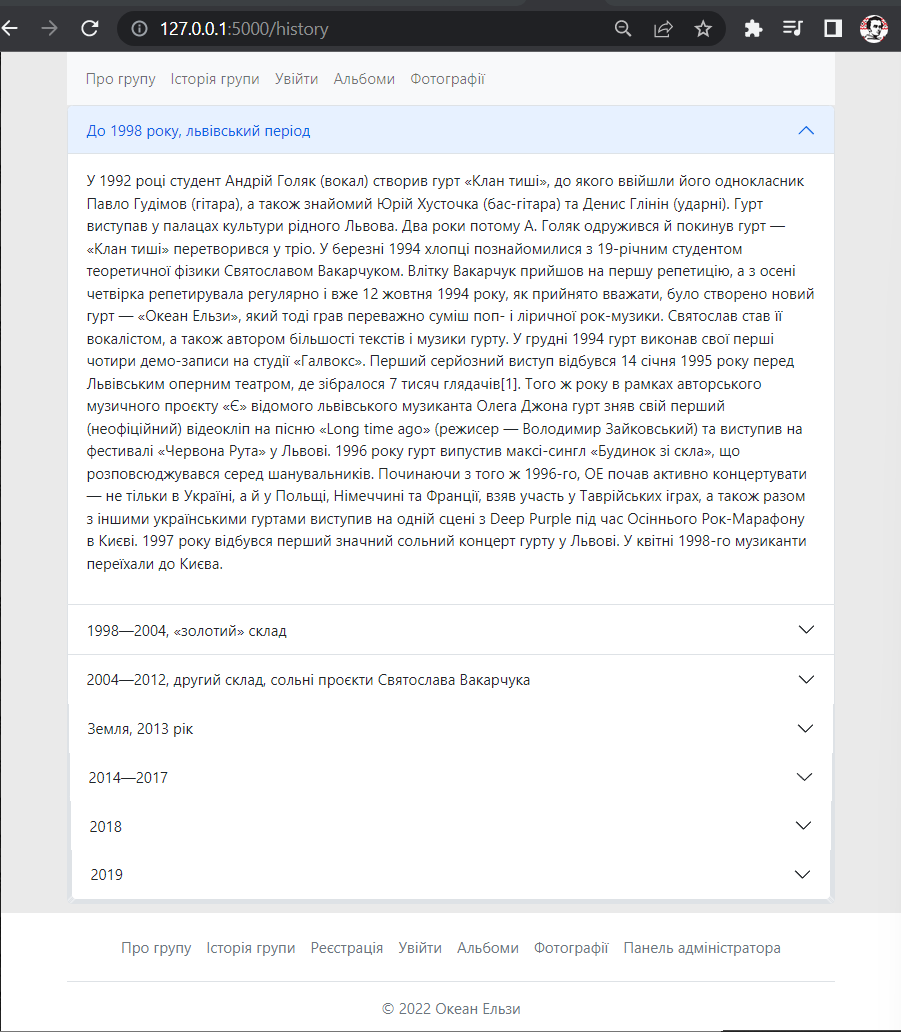


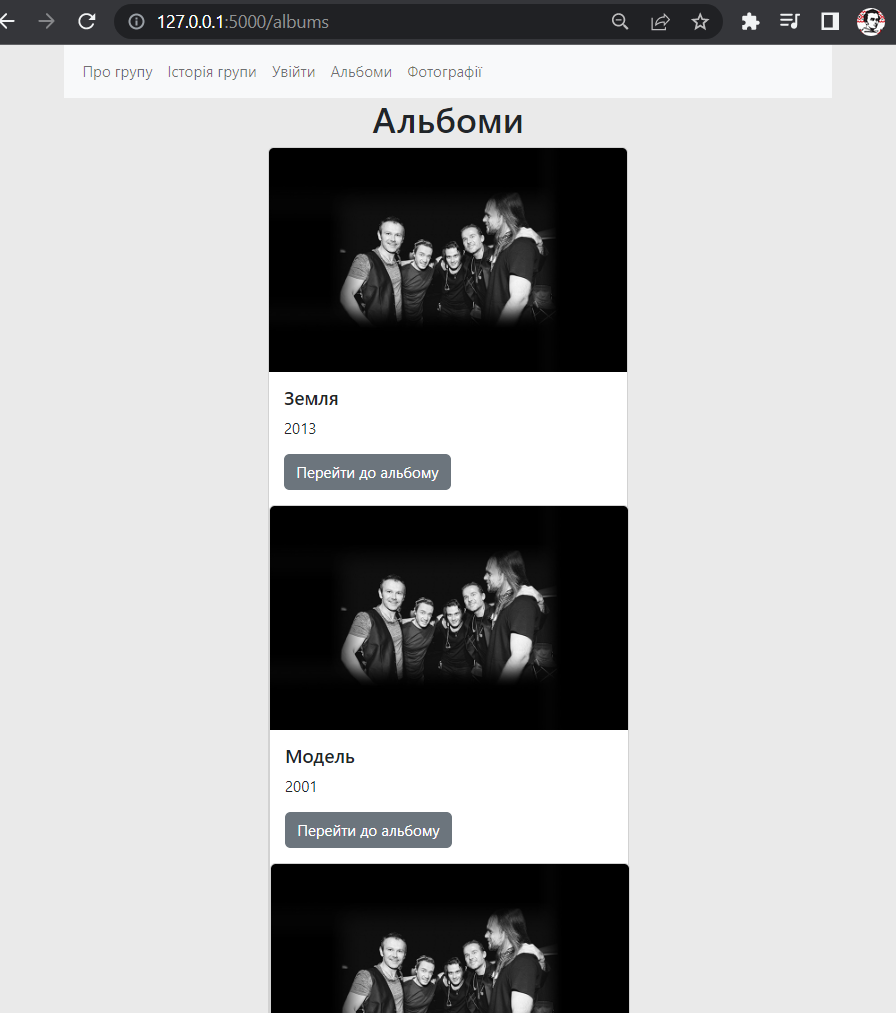


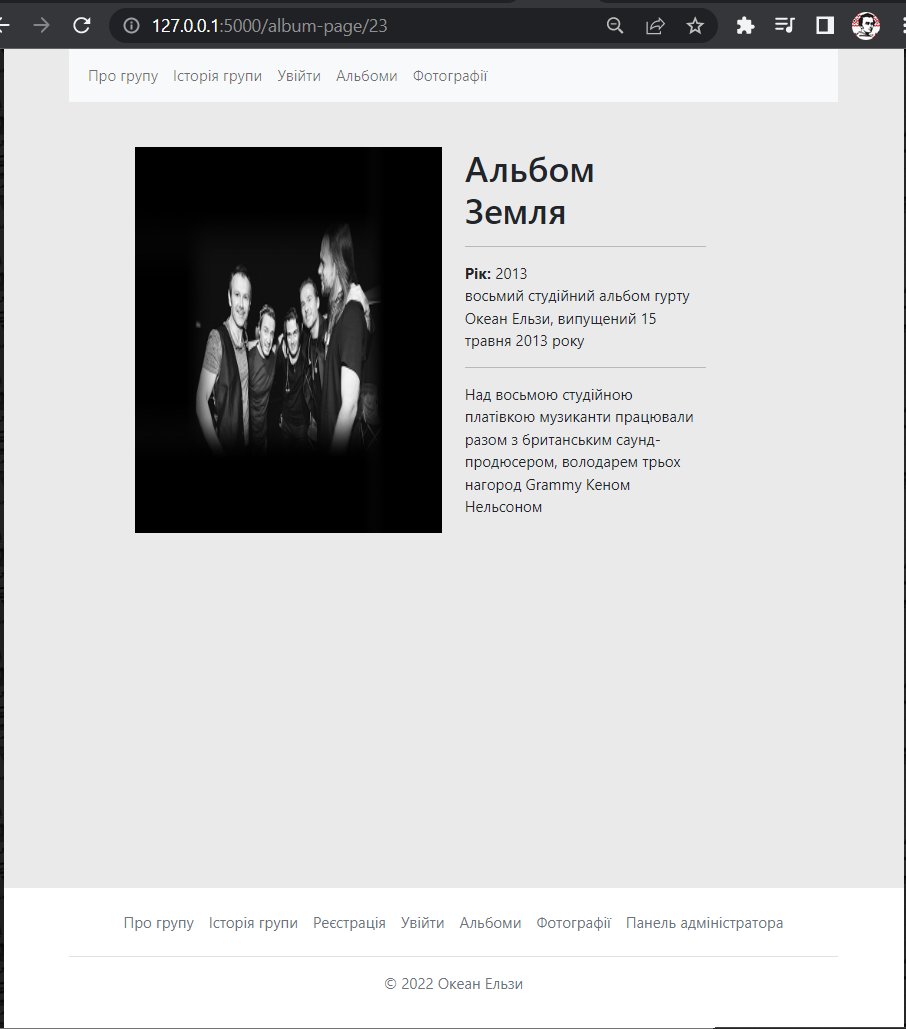


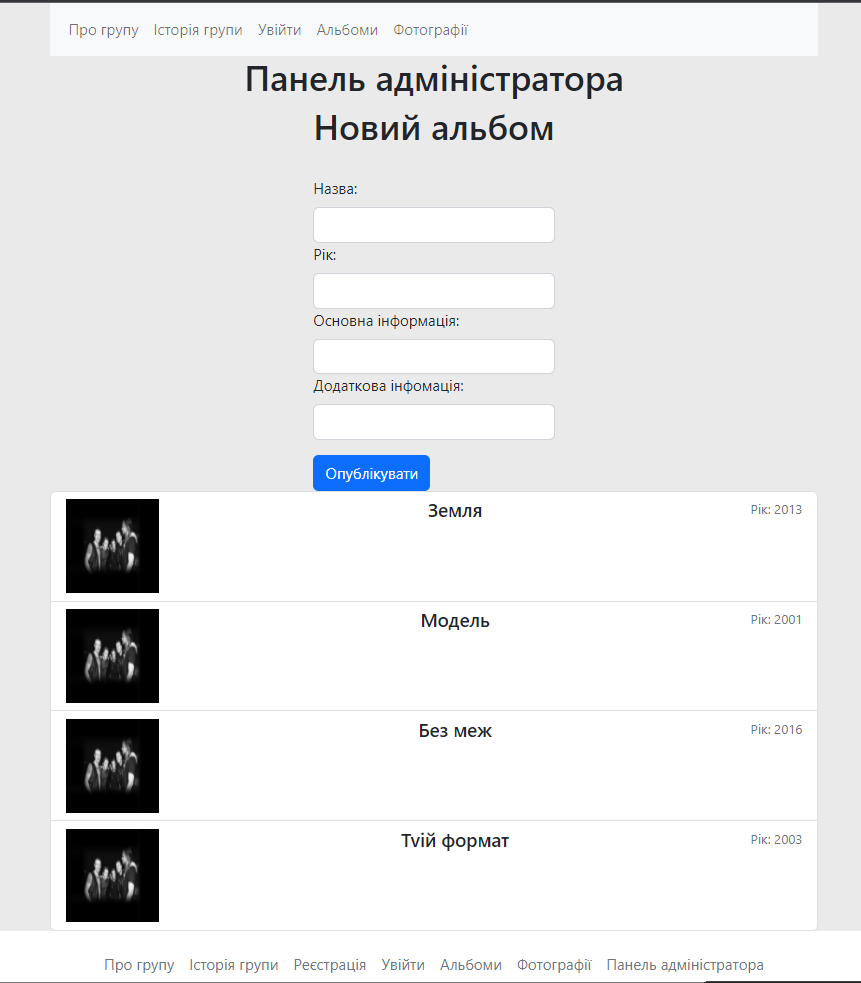
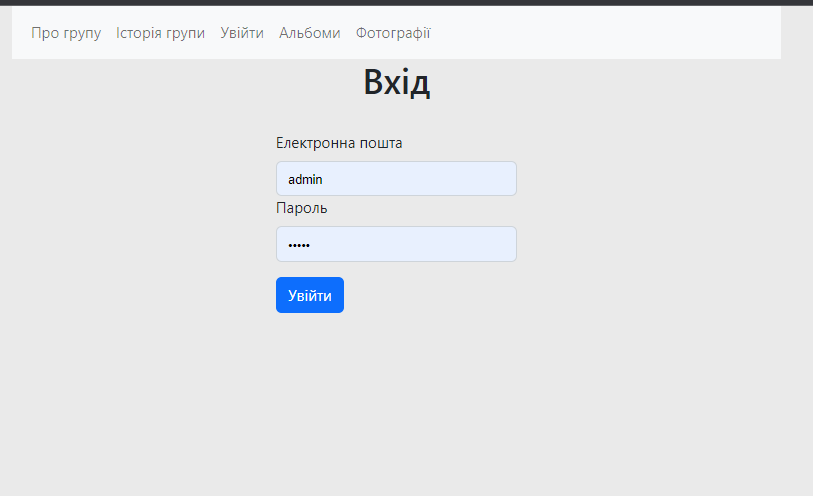
*import* telebot  
*from* telebot *import* types  
*import* random  
*from* datetime *import* date  
*import* time  
  
TOKEN = "5949276147:AAEEhfpS-CMTbGWmO2XJCZDLjtQ46K1gzD4"  
bot = telebot.TeleBot(TOKEN)  
RAND = 0  
USED = []  
ANSWER = 0  
START\_TIME = 0  
test = [  
 {  
 'question': 'Коли народився Шевченко Т.Г.?',  
 'a': 1814,  
 'b': 1818,  
 'c': 1826,  
 'd': 1817,  
 'answer': 1814  
 },  
 {  
 'question': 'Коли був виданий "Кобзар"?',  
 'a': 1834,  
 'b': 1840,  
 'c': 1846,  
 'd': 1847,  
 'answer': 1840  
 },  
 {  
 'question': 'Який був перший твір Шевченка?',  
 'a': 'Тарасова ніч',  
 'b': 'Причинна',  
 'c': 'Катерина',  
 'd': 'Перебендя',  
 'answer': 'Причинна'  
 }  
]  
  
  
@bot.message\_handler(commands=['start'])  
*def* start(message):  
 *global* RAND  
 *global* USED  
 *global* ANSWER  
 *global* START\_TIME  
 START\_TIME = time.time()  
 ANSWER = 0  
 USED.clear()  
 RAND = random.randint(0,2)  
 USED.append(RAND)  
 keyboard = types.InlineKeyboardMarkup(row\_width=2)  
 btn1 = types.InlineKeyboardButton(test[RAND]['a'], callback\_data=test[RAND]['a'])  
 btn2 = types.InlineKeyboardButton(test[RAND]['b'], callback\_data=test[RAND]['b'])  
 btn3 = types.InlineKeyboardButton(test[RAND]['c'], callback\_data=test[RAND]['c'])  
 btn4 = types.InlineKeyboardButton(test[RAND]['d'], callback\_data=test[RAND]['d'])  
  
 keyboard.add(btn1, btn2, btn3, btn4)  
 bot.send\_message(  
 message.chat.id,  
 test[RAND]['question'],  
  
 reply\_markup=keyboard  
 )  
  
  
@bot.callback\_query\_handler(func=*lambda* callback: callback.data)  
*def* check\_callback\_data(callback):  
 *global* ANSWER  
 *global* RAND  
 *global* USED  
 print(f'callback => {callback.data}')  
 print(f'REND => {RAND}')  
 *if* callback.data == str(test[RAND]['answer']):  
 print('right')  
 ANSWER += 1  
 bot.edit\_message\_text(  
 chat\_id=callback.message.chat.id,  
 message\_id=callback.message.id,  
 text='Шевченко пишається тобою',  
 reply\_markup=*None* )  
 *else*:  
 bot.edit\_message\_text(  
 chat\_id=callback.message.chat.id,  
 message\_id=callback.message.id,  
 text='Шевченко обурений',  
 reply\_markup=*None* )  
 *while* RAND *in* USED *and* len(USED) < 3:  
 RAND = random.randint(0, 2)  
  
 USED.append(RAND)  
 *if* len(USED) < 4:  
 keyboard = types.InlineKeyboardMarkup(row\_width=2)  
 btn1 = types.InlineKeyboardButton(test[RAND]['a'], callback\_data=test[RAND]['a'])  
 btn2 = types.InlineKeyboardButton(test[RAND]['b'], callback\_data=test[RAND]['b'])  
 btn3 = types.InlineKeyboardButton(test[RAND]['c'], callback\_data=test[RAND]['c'])  
 btn4 = types.InlineKeyboardButton(test[RAND]['d'], callback\_data=test[RAND]['d'])  
 keyboard.add(btn1, btn2, btn3, btn4)  
 bot.send\_message(  
 callback.message.chat.id,  
 test[RAND]['question'],  
 reply\_markup=keyboard  
 )  
 *else*:  
 *global* START\_TIME  
 today = date.today()  
 bot.send\_message(  
 callback.message.chat.id,  
 f"Результат {ANSWER}/3\nДата проходження =>{today.strftime('%m/%d/%y')}\nЧас проходження =>{int(time.time()-START\_TIME)}c"  
 )  
  
bot.polling(none\_stop=*True*)











Code:

Base.html

<!DOCTYPE *html*>  
<html *lang*="en">  
<head>  
 <meta *charset*="UTF-8">  
 <title>{% *block* head %}{% *endblock* %}</title>  
 <link *href*="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" *rel*="stylesheet"  
 *integrity*="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65" *crossorigin*="anonymous">  
 <link *rel*="stylesheet" *href*="https://cdn.jsdelivr.net/npm/@fancyapps/ui@4.0/dist/fancybox.css"/>  
 <link *rel*="stylesheet" *href*="{{ url\_for('static',filename = 'style.css') }}">  
 <script *src*="https://cdn.jsdelivr.net/npm/@fancyapps/ui@4.0/dist/fancybox.umd.js"></script>  
</head>  
<body>  
<div *class*="container main-container">  
 <nav *class*="navbar navbar-expand-lg bg-light">  
 <div *class*="container-fluid">  
 <button *class*="navbar-toggler" *type*="button" *data-bs-toggle*="collapse" *data-bs-target*="#navbarNav"  
 *aria-controls*="navbarNav" *aria-expanded*="false" *aria-label*="Toggle navigation">  
 <span *class*="navbar-toggler-icon"></span>  
 </button>  
 <div *class*="collapse navbar-collapse" *id*="navbarNav">  
 <ul *class*="navbar-nav">  
 <li *class*="nav-item">  
 <a *class*="nav-link" *href*="/">Про групу</a>  
 </li>  
 <li *class*="nav-item">  
 <a *class*="nav-link" *href*="/history">Історія групи</a>  
 </li>  
 <li *class*="nav-item">  
 <a *class*="nav-link" *href*="/login">Увійти</a>  
 </li>  
 <li *class*="nav-item">  
 <a *class*="nav-link" *href*="/albums">Альбоми</a>  
 </li>  
 <li *class*="nav-item">  
 <a *class*="nav-link" *href*="/photos">Фотографії</a>  
 </li>  
 </ul>  
 </div>  
 </div>  
 </nav>  
  
 {% *block* body %}{% *endblock* %}  
</div>  
<div *class*="container footer">  
 <footer *class*="py-3">  
 <ul *class*="nav justify-content-center border-bottom pb-3 mb-3">  
 <li *class*="nav-item"><a *href*="/" *class*="nav-link px-2 text-muted">Про групу</a></li>  
 <li *class*="nav-item"><a *href*="/history" *class*="nav-link px-2 text-muted">Історія групи</a></li>  
 <li *class*="nav-item"><a *href*="/signin" *class*="nav-link px-2 text-muted">Реєстрація</a></li>  
 <li *class*="nav-item"><a *href*="/login" *class*="nav-link px-2 text-muted">Увійти</a></li>  
 <li *class*="nav-item"><a *href*="/albums" *class*="nav-link px-2 text-muted">Альбоми</a></li>  
 <li *class*="nav-item"><a *href*="/photos" *class*="nav-link px-2 text-muted">Фотографії</a></li>  
 <li *class*="nav-item"><a *href*="/admin-page" *class*="nav-link px-2 text-muted">Панель адміністратора</a></li>  
 </ul>  
 <p *class*="text-center text-muted">© 2022 Океан Ельзи</p>  
 </footer>  
</div>  
  
  
<script *src*="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"  
 *integrity*="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcpxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4"  
 *crossorigin*="anonymous"></script>  
</body>  
</html>

Адмін сторінка з обробкою даних

{% *extends* 'base.html' %}  
  
{% *block* head %}  
 Панель адміністратора  
{% *endblock* %}  
  
{% *block* body %}  
 <h1 *class*="text-center">Панель адміністратора</h1>  
  
  
 <h1 *class*="text-center">Новий альбом</h1>  
  
 <form *action*="{{url\_for('admin\_page')}}" *method*="post" *class*="row g-3 mt-3">  
 <div *class*="contatiner">  
 <div *class*="row">  
 <div *class*="col-4"></div>  
 <div *class*="col-4">  
 <div *class*="col-12">  
 <label *for*="inputEmail4" *class*="form-label">Назва:</label>  
 <input *type*="text" *name*="name" *class*="form-control" *id*="inputEmail4">  
 </div>  
 <div *class*="col-12">  
 <label *for*="inputPassword4" *class*="form-label">Рік:</label>  
 <input *type*="number" *name*="year" *class*="form-control" *id*="inputPassword4">  
 </div>  
 <div *class*="col-12">  
 <label *for*="inputPassword4" *class*="form-label">Основна інформація:</label>  
 <input *type*="text" *name*="info" *class*="form-control" *id*="inputPassword4">  
 </div>  
 <div *class*="col-12">  
 <label *for*="inputPassword4" *class*="form-label">Додаткова інфомація:</label>  
 <input *type*="text" *name*="info2" *class*="form-control" *id*="inputPassword4">  
 </div>  
 <div *class*="col-12 mt-3">  
 <button *type*="submit" *class*="btn btn-primary">Опублікувати</button>  
 </div>  
 {% *if* message %}  
 <h6 *style*="color: red">{{ message }}</h6>  
 {% *endif* %}  
 </div>  
 <div *class*="col-4"></div>  
 </div>  
 </div>  
  
 </form>  
 <div *class*="list-group">  
 {% *for* p *in* post %}  
 <a *href*="/edit-page/{{ p[0] }}" *class*="list-group-item list-group-item-action">  
 <div *class*="d-flex w-100 justify-content-between">  
 <img *class*="admin-page-album-img" *src*="{{ url\_for('static',filename = 'img/slider-1.jpg') }}" *alt*="">  
 <h5 *class*="mb-1">{{ p[1] }}</h5>  
 <small *class*="text-muted">Рік: {{ p[2] }}</small>  
 </div>  
 </a>  
 {% *endfor* %}  
 </div>  
  
{% *endblock* %}

Routing + підключення бази даних

*from* flask *import* Flask, flash, render\_template, redirect, request, g, abort  
*import* sqlite3  
*import* os  
  
DATABASE = '/tmp/flsite.db'  
DEBUG = *True*SECRET\_KEY = 'di&l594&$89mfh$&h$g4'  
  
app = Flask(\_\_name\_\_)  
app.config.from\_object(\_\_name\_\_)  
  
app.config.update(dict(DATABASE=os.path.join(app.root\_path,"lfsite.db")))  
  
*def* connect\_db():  
 conn = sqlite3.connect(app.config['DATABASE'])  
 conn.row\_factory = sqlite3.Row  
 *return* conn  
  
  
*def* create\_db():  
 db = connect\_db()  
 *with* app.open\_resource('sq\_db.sql', mode='r') *as* f:  
 db.cursor().executescript(f.read())  
 db.commit()  
 db.close()  
  
  
*def* get\_db():  
 *if not* hasattr(g, 'link\_db'):  
 g.link\_db = connect\_db()  
 *return* g.link\_db  
  
  
@app.teardown\_appcontext  
*def* close\_db(error):  
 *if* hasattr(g, 'link\_db'):  
 g.link\_db.close()  
  
*from* FDataBase *import* FDataBase  
*# Клас для роботи з бд*@app.route('/')  
*def* index():  
 *return* render\_template("index.html")  
  
  
  
  
@app.route("/albums", methods=["POST", "GET"])  
*def* showPost():  
 db = get\_db()  
 dbase = FDataBase(db)  
 post = dbase.getAllPost()  
  
  
 *return* render\_template('albums.html', post=post)  
  
  
@app.errorhandler(404)  
*def* page\_not\_found(e):  
 *return* render\_template('404.html'), 404  
  
  
@app.route('/album-page/<int:id>')  
*def* album\_page(id):  
 db = get\_db()  
 dbase = FDataBase(db)  
 post = dbase.getPost(id)  
 *return* render\_template('album-page.html', post=post)  
  
  
@app.route('/admin-page', methods=["POST", "GET"])  
*def* admin\_page():  
 db = get\_db()  
 dbase = FDataBase(db)  
 post = dbase.getAllPost()  
  
 *if* request.method == "POST":  
 *if* len(request.form['name'])>1:  
 res = dbase.AddAlbum(request.form['name'], request.form['year'], request.form['info'], request.form['info2'])  
 *if not* res:  
 flash('Помилка додавання альбому', category="error")  
 *else*:  
 flash('Альбом додано успішно', category='success')  
 *else*:  
 flash('Помилка додавання альбому', category='error')  
  
  
 *return* render\_template('admin-page.html', post=post)  
  
  
@app.route('/edit-page/<int:id>', methods=["POST", "GET"])  
*def* edit\_page(id):  
 db = get\_db()  
 dbase = FDataBase(db)  
 post = dbase.getPost(id)  
 *if* request.method == "POST":  
 *if* len(request.form['name'])>1:  
 res = dbase.updatePost(id, request.form['name'], request.form['year'], request.form['info'], request.form['info2'])  
 *if not* res:  
 flash('Помилка додавання альбому', category="error")  
 *else*:  
 flash('Альбом додано успішно', category='success')  
 *else*:  
 flash('Помилка додавання альбому', category='error')  
 *return* render\_template('edit-page.html', post=post)  
  
  
  
@app.route('/history')  
*def* history():  
 *return* render\_template('history.html')  
  
  
  
  
@app.route('/login', methods=["POST", "GET"])  
*def* login():  
 message = ""  
 *if* request.method == "POST":  
 *#Сторінка аміна тільки адміну  
 if* request.form['name'] == 'admin' *and* request.form['password'] == 'admin':  
 *return* redirect("/admin-page")  
 message = "Ви не адмін"  
 *return* render\_template('login.html', message=message)  
  
  
@app.route('/photos')  
*def* photos():  
 example\_photos = ['slider-2.jpg', 'slider-1.jpg', 'slider-2.jpg', 'slider-3.jpg', 'slider-2.jpg']  
 *return* render\_template('photos.html', photos=example\_photos)  
  
  
  
  
*if* \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=*True*)

Клас для роботи з базою даних

*import* sqlite3  
*import* json  
  
*class* FDataBase:  
 *def* \_\_init\_\_(*self*, db):  
 *self*.\_\_db = db  
 *self*.\_\_cur = db.cursor()  
  
 *# додавання нового альбрму  
 def* AddAlbum(*self*, name, year, info, info2):  
 *try*:  
 *self*.\_\_cur.execute("INSERT INTO albums VALUES(NULL, ?, ?, ?, ?)", (name, year, info, info2))  
 *self*.\_\_db.commit()  
 *except* sqlite3.Error *as* e:  
 print("Помилка дадавання в БД"+str(e))  
 *return False  
 return True  
  
 #Отримати окремий альбом  
 def* getPost(*self*, postId):  
 *try*:  
 *self*.\_\_cur.execute(f"SELECT \* FROM albums WHERE id = {postId} LIMIT 1")  
 res = *self*.\_\_cur.fetchone()  
 *if* res:  
 *return* res  
 *except* sqlite3.Error *as* e:  
 print("Помилка отримання даних з БД" + str(e))  
  
 *return* (*False*, *False*)  
  
 *# Отримати всі альбоми  
 def* getAllPost(*self*):  
 *try*:  
 *self*.\_\_cur.execute(f"SELECT \* FROM albums")  
 res = *self*.\_\_cur.fetchall()  
 *if* res:  
 *return* res  
 *except* sqlite3.Error *as* e:  
 print("Помилка отримання даних з БД" + str(e))  
  
 *return* (*False*, *False*)  
  
 *#Оновити альбом  
 def* updatePost(*self*, postId, name, year, info, info2):  
 *try*:  
 print(name)  
 print(postId)  
 *self*.\_\_cur.execute(f"UPDATE albums SET name = ?, year = ?, info = ?, info2 = ? WHERE id LIKE {postId}", (name, year, info, info2))  
 *self*.\_\_db.commit()  
 *except* sqlite3.Error *as* e:  
 print("Помилка отримання даних з БД" + str(e))  
  
 *return* (*False*, *False*)  
  
 *def* deletePost(*self*, postId):  
 *try*:  
 *self*.\_\_cur.execute(f"DELETE FROM albums WHERE id LIKE {postId}")  
 *self*.\_\_db.commit()  
 *except* sqlite3.Error *as* e:  
 print("Помилка видалення даних з БД" + str(e))  
  
 *return* (*False*, *False*)