ДОДАТОК А

Основний код програми

```python
from flask import Flask, render_template, redirect,
url_for, flash, abort, session, request
from functools import wraps
from flask_bootstrap import Bootstrap
from flask_ckeditor import CKEditor
from datetime import date
from werkzeug.security import generate_password_hash,
check_password_hash
from flask_sqlalchemy import SQLAlchemy
from sqlalchemy.orm import relationship
from flask_login import UserMixin, login_user,
LoginManager, login_required, current_user, logout_user
from forms import CreatePostForm, NewUserForm, Login,
CreateQuestion, START, GiveAnswer, END
from flask_gravatar import Gravatar

app = Flask(__name__)
app.config['SECRET_KEY'] =
'8BYkEfBA6O6donzWlSihBXox7C0sKR6b'
Bootstrap(app)
login_manager = LoginManager()
login_manager.init_app(app)


@login_manager.user_loader
def load_user(user_id):
    return User.query.get(int(user_id))


##CONNECT TO DB
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///
blog.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
db = SQLAlchemy(app)


##CONFIGURE TABLES


class User(UserMixin, db.Model):
```

```python
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String, nullable=False)
    email = db.Column(db.String, nullable=False,
unique=True)
    password = db.Column(db.String, nullable=False)
    user = db.relationship('BlogPost', backref='user')

class BlogPost(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    author = db.Column(db.String(250), nullable=False)
    title = db.Column(db.String(250), unique=True,
nullable=False)
    subtitle = db.Column(db.String(250), nullable=False)
    date = db.Column(db.String(250), nullable=False)
    img_url = db.Column(db.String(250), nullable=False)
    user_id = db.Column(db.Integer,
db.ForeignKey('user.id'))

class Questions(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    blog_id = db.Column(db.Integer,
db.ForeignKey('blog_post.id'))
    text = db.Column(db.Text, nullable=False)

class Answers(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String, nullable=False)
    blog_id = db.Column(db.Integer, nullable=False)
    text_questions = db.Column(db.Text, nullable=False)
    text_answer = db.Column(db.Text, nullable=False)

with app.app_context():
    db.create_all()

def admin_only(function):
    @wraps(function)
    def wrapper(*args, **kwargs):
        if not '_user_id' in session.keys():
            flash('Page locked, login as admin.')
            return redirect(url_for('login'))
        if int(session['_user_id']) == 1:
            return function(*args, **kwargs)
        else:
            abort(403)
    return wrapper
```

```python
@app.route('/')
def get_all_posts():
    posts = BlogPost.query.all()
    return render_template("index.html", all_posts=posts,
logged=current_user.is_authenticated)


@app.route('/check/<int:post_id>')
@admin_only
def check_table(post_id):
    questions = Answers.query.filter_by(blog_id=post_id)
    names = set()
    for question in questions:
        names.add(question.name)
    return render_template('check_answ.html',
logged=current_user.is_authenticated, names=names,
post_id=post_id, users=User())


@app.route('/check/<int:post_id>/<int:user_id>')
def check(post_id, user_id):
    user = User.query.filter_by(id=user_id).first()
    answer_user = Answers.query.filter_by(name=user.name)
    answer_post = []
    for x in answer_user:
        if x.blog_id == post_id:
            answer_post.append(x)
    print(answer_post)
    return render_template('check.html',
logged=current_user.is_authenticated, user=user,
answer=answer_post)


@app.route('/register', methods=['GET', 'POST'])
def register():
    form = NewUserForm()
    if form.validate_on_submit():
        email = form.email.data
        if User.query.filter_by(email=email).first():
            flash('You are already registered with this
email. Come in.')
            return redirect(url_for('login'))
        new_user = User(
            name=form.name.data,
            email=email,
```

```python
                                    password=generate_password_hash(form.password.data,
salt_length=8))
        db.session.add(new_user)
        db.session.commit()
        return redirect(url_for('login'))
    return render_template("register.html", form=form,
logged=current_user.is_authenticated)


@app.route('/login', methods=['GET', 'POST'])
def login():
    form = Login()
    user =
User.query.filter_by(email=form.email.data).first()
    if form.is_submitted():
        if user:
            if check_password_hash(user.password,
form.password.data):
                login_user(user)
                return redirect(url_for('get_all_posts'))
            else:
                flash('Не вірний email або пароль')
                return redirect(url_for('login'))
        else:
            flash('Користувач не знайдений')
            return redirect(url_for('login'))
    return render_template("login.html", form=form,
logged=current_user.is_authenticated)


@app.route('/logout')
def logout():
    logout_user()
    return redirect(url_for('login'))


@app.route("/post/<int:post_id>", methods=['GET',
'POST'])
def show_post(post_id):
    if not current_user.is_authenticated:
        flash('Спочатку увійди')
        return redirect(url_for('login'))
    length_questions =
len(Questions.query.filter_by(blog_id=post_id).all())
```

```python
    form = START()
    if form.validate_on_submit():
        return redirect(url_for('give_answers',
post_id=post_id, question_id=0,
logged=current_user.is_authenticated))
    requested_post = BlogPost.query.get(post_id)
    return render_template("post.html",
post=requested_post, form=form, len=length_questions,
l=current_user, logged=current_user.is_authenticated)


@app.route("/post/<int:post_id>/<int:question_id>",
methods=['GET', 'POST'])
def give_answers(post_id, question_id):
    questions =
Questions.query.filter_by(blog_id=post_id).all()
    if question_id+1 > len(questions):
        return redirect(url_for('end'))
    question = questions[question_id]
    form = GiveAnswer()
    if form.validate_on_submit():
        print(post_id)
        new_answer = Answers(name=current_user.name,

text_questions=question.text,
                             blog_id=post_id,
                             text_answer=form.answer.data
                             )
        db.session.add(new_answer)
        db.session.commit()
        return redirect(url_for('give_answers',
post_id=post_id, question_id=question_id+1))
    print(question)
    return render_template('questions.html', form=form,
logged=current_user.is_authenticated, question=question,
                           N=question_id+1)


@app.route('/post/end', methods=['GET', 'POST'])
def end():
    form = END()
    if form.validate_on_submit():
        return redirect(url_for('get_all_posts'))
    return render_template('End.html', form=form,
logged=current_user.is_authenticated)
```

```python
@app.route("/new-post", methods=['GET', 'POST'])
@admin_only
def add_new_post():
    form = CreatePostForm()
    if form.validate_on_submit():
        new_post = BlogPost(
            title=form.title.data,
            subtitle=form.subtitle.data,
            img_url=form.img_url.data,
            author=current_user.name,
            date=date.today().strftime("%B %d, %Y"),
            user_id=current_user.id
        )
        db.session.add(new_post)
        db.session.commit()
        return redirect(url_for("add_new_question"))
    return render_template("make-post.html", form=form,
logged=current_user.is_authenticated)


@app.route("/new-question", methods=['GET', 'POST'])
@admin_only
def add_new_question():
    form = CreateQuestion()
    if 'chancel' in request.form:
        form.question.validators = []
        return redirect(url_for('get_all_posts',
logged=current_user.is_authenticated))
    if form.validate_on_submit():
            last_blog_post =
BlogPost.query.order_by(BlogPost.id.desc()).first()
            new_question = Questions(
                text=form.question.data,
                blog_id=last_blog_post.id
            )
            db.session.add(new_question)
            db.session.commit()
            return redirect(url_for('add_new_question',
form=form, logged=current_user.is_authenticated))
    return render_template("create_question.html",
form=form, logged=current_user.is_authenticated)




@app.route("/delete/<int:post_id>")
```

```python
@admin_only
def delete_post(post_id):
    post_to_delete = BlogPost.query.get(post_id)
    question =
Questions.query.filter_by(blog_id=post_id).all()
    for q in question:
        db.session.delete(q)
    db.session.delete(post_to_delete)
    db.session.commit()
    return redirect(url_for('get_all_posts'))


if __name__ == "__main__":
    app.run(debug=True)
```

HTML-код головної стрінки

```html
{% include "header.html" %}

  <!-- Page Header -->
  <header class="masthead" style="height: 480px; background-image:
url('https://images.unsplash.com/photo-1470092306007-055b6797ca72?
ixlib=rb-1.2.1&auto=format&fit=crop&w=668&q=80')">
    <div class="overlay"></div>
    <div class="container">
     <div class="row">
      <div class="col-lg-8 col-md-10 mx-auto">
       <div class="site-heading">
        <h1>Колекції Питань</h1>
        <span class="subheading"></span>
       </div>
      </div>
     </div>
    </div>
  </header>

  <!-- Main Content -->
  <div class="container">
   <div class="row">
     <div class="col-lg-8 col-md-10 mx-auto">
      {% for post in all_posts %}
      <div class="post-preview">
       <a href="{{ url_for('show_post', post_id=post.id) }}">
        <h2 class="post-title">
         {{post.title}}
        </h2>
```

```html
        <h3 class="post-subtitle">
          {{post.subtitle}}
        </h3>
      </a>
      <p class="post-meta">Posted by
        <a href="#">{{post.author}}</a>
        on {{post.date}}
        {% if current_user.id == 1 %}
          <a href="{{url_for('delete_post', post_id=post.id) }}">✘</a>
        {% endif %}
      </p>
    </div>
    <hr>
    {% endfor %}

    <!-- New Post -->
    {% if current_user.id == 1 %}
      <div class="clearfix">
        <a class="btn btn-primary float-right"
href="{{url_for('add_new_post')}}">Створити нову колекцію питань</a>
      </div>
    {% endif %}
   </div>
  </div>
 </div>
 <hr>

{% include "footer.html" %}
```

Html-код header

```html
<!DOCTYPE html>
<html lang="en">

<head>

  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
  <meta name="description" content="">
  <meta name="author" content="">

  <title>Мій Опитувальник</title>

  <!-- Bootstrap core CSS -->
  <link href="{{ url_for('static', filename='vendor/bootstrap/css/
bootstrap.min.css') }}" rel="stylesheet">
```

```html
    <!-- Custom fonts for this template -->
    <link href="{{ url_for('static', filename='vendor/fontawesome-free/css/all.min.css')}}" rel="stylesheet" type="text/css">
    <link href='https://fonts.googleapis.com/css?family=Lora:400,700,400italic,700italic' rel='stylesheet' type='text/css'>
    <link href='https://fonts.googleapis.com/css?family=Open+Sans:300italic,400italic,600italic,700italic,800italic,400,300,600,700,800' rel='stylesheet' type='text/css'>

    <!-- Custom styles for this template -->
    <link href="{{ url_for('static', filename='css/clean-blog.min.css')}}" rel="stylesheet">

</head>

<body>

    <!-- Navigation -->
    <nav class="navbar navbar-expand-lg navbar-light fixed-top" id="mainNav">
      <div class="container">
        <a class="navbar-brand" href="{{url_for('get_all_posts')}}">
        </a>
        <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse" data-target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation">
          Menu
          <i class="fas fa-bars"></i>
        </button>
        <div class="collapse navbar-collapse" id="navbarResponsive">
          <ul class="navbar-nav ml-auto">
            <li class="nav-item">
              <a class="nav-link" href="{{ url_for('get_all_posts') }}">Дім</a>
            </li>
            {% if not logged %}
            <li class="nav-item">
              <a class="nav-link" href="{{ url_for('login') }}">Вхід</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="{{ url_for('register') }}">Реєстрація</a>
            </li>
            {% endif %}
            {% if logged %}
```

```html
        <li class="nav-item">
          <a class="nav-link" href="{{ url_for('logout') }}">Вихід</a>
        </li>
        {% endif %}
      </ul>
    </div>
  </div>
</nav>
```

Html-код footer
```html
  <!-- Footer -->
  <footer>
    <div class="container">
      <div class="row">
        <div class="col-lg-8 col-md-10 mx-auto">
          <ul class="list-inline text-center">
            <li class="list-inline-item">
              <a href="#">
                <span class="fa-stack fa-lg">
                  <i class="fas fa-circle fa-stack-2x"></i>
                  <i class="fab fa-twitter fa-stack-1x fa-inverse"></i>
                </span>
              </a>
            </li>
            <li class="list-inline-item">
              <a href="#">
                <span class="fa-stack fa-lg">
                  <i class="fas fa-circle fa-stack-2x"></i>
                  <i class="fab fa-facebook-f fa-stack-1x fa-inverse"></i>
                </span>
              </a>
            </li>
            <li class="list-inline-item">
              <a href="#">
                <span class="fa-stack fa-lg">
                  <i class="fas fa-circle fa-stack-2x"></i>
                  <i class="fab fa-github fa-stack-1x fa-inverse"></i>
                </span>
              </a>
            </li>
          </ul>
          <p class="copyright text-muted">Copyright &copy; Your Website
2023</p>
        </div>
      </div>
```

```html
    </div>
  </footer>

  <!-- Bootstrap core JavaScript -->
  <script src="{{ url_for('static', filename='vendor/jquery/jquery.min.js')}}"></script>
  <script src="{{ url_for('static', filename='vendor/bootstrap/js/bootstrap.bundle.min.js')}}"></script>

  <!-- Custom scripts for this template -->
  <script src="{{ url_for('static', filename='js/clean-blog.min.js')}}"></script>

</body>

</html>
```