FLASK WEB DEVELOPMENT (URL SHORTENER)

Why Do We Need a URL Shortener?

Sometimes we need to share or send links, and this can be tiresome and annoying to copy and paste long URLs. That is where URL shorteners come in. Not only it helps in shortening the URL but it also allows the user to copy the shortened URL with a click of a button.

The Project Consists of 3 Parts:

1. Frontend (Done With HTML, CSS, and Bootstrap)
2. Backend - Flask (Python)
3. Backend - Database ORM

1. Importing Necessary Modules

```
# Import Necessary Libraries

import os
import random
import string
from flask import Flask, render_template, request, url_for, redirect
from flask_sqlalchemy import SQLAlchemy
from flask_migrate import Migrate
```

2. Initialize Name

```
# Initialize Name

app = Flask(__name__)
```

3. **SQL Alchemy Configuration**

```
# SQL Alchemy Configuration

basedir = os.path.abspath(os.path.dirname(__file__))
path = 'sqlite:///' + os.path.join(basedir, 'data.db')
app.config['SQLALCHEMY_DATABASE_URI'] = path
app.config['SQLALCHEMY_TRACK_MODIFICATION'] = False

db = SQLAlchemy(app)
Migrate(app,db)
```

4. **Create a Model**

```
# Create a Model

class urls(db.Model):
    id = db.Column(db.Integer, primary_key = True)
    original = db.Column(db.String())
    short = db.Column(db.String(15))

    def __init__(self, original, short):
        self.original = original
        self.short = short

    def __repr__(self) -> str:
        return f"{self.original} - {self.short}"

def shorten_url():
    total_characters = string.ascii_lowercase + string.ascii_uppercase + string.digits
    while True:
        random_char = random.choices(total_characters, k = 7)
        random_char = "".join(random_char)
        short_url = urls.query.filter_by(short = random_char).first()
        if not short_url:
            return random_char
```

5. Create End Points for Backend

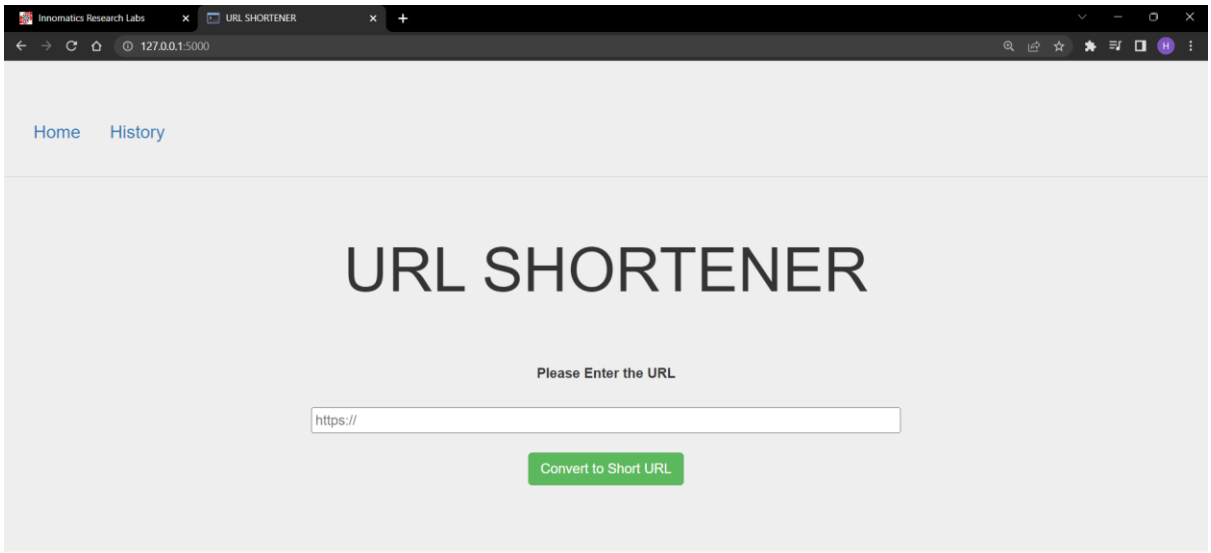
```python
# Create end points for backend

@app.route('/', methods=['POST', 'GET'])
def home():
    if request.method == "POST":
        url_got = request.form["url_link"]
        url_found = urls.query.filter_by(original = url_got).first()

        if url_found:
            return redirect(url_for("display_short_url", url = url_found.short))
        else:
            short_url = shorten_url()
            new_url = urls(url_got, short_url)
            db.session.add(new_url)
            db.session.commit()
            return redirect(url_for("display_short_url", url = short_url))
    else:
        return render_template('url_Page.html')

@app.route('/<short_url>')
def redirecting(short_url):
    original_url = urls.query.filter_by(short = short_url).first()
    if original_url:
        return redirect(original_url.original)
    else:
        return f'<h1>Url Does Not Exist!</h1>'
```

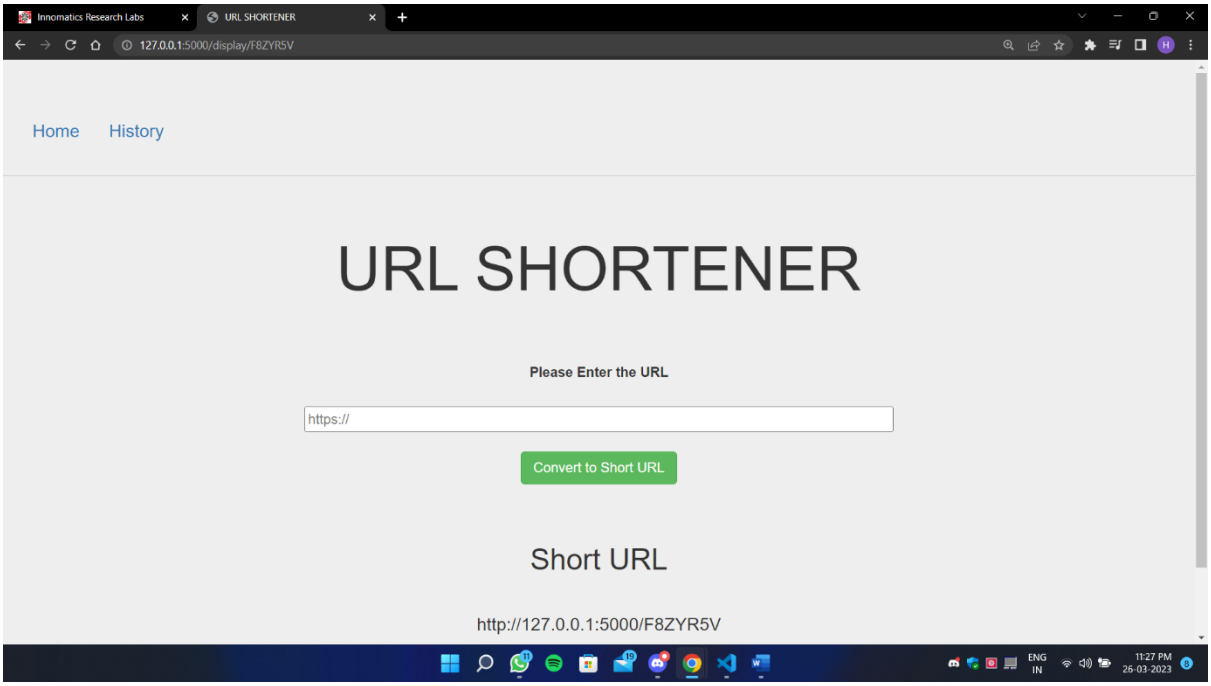
```python
@app.route('/<short_url>')
def redirecting(short_url):
    original_url = urls.query.filter_by(short = short_url).first()
    if original_url:
        return redirect(original_url.original)
    else:
        return f'<h1>Url Does Not Exist!</h1>'

@app.route('/display/<url>')
def display_short_url(url):
    return render_template('url_Page.html', short_url_display = url)

@app.route('/delete/<int:id>')
def delete(id):
    url = urls.query.filter_by(id = id).first()
    db.session.delete(url)
    db.session.commit()
    return redirect("/history")

@app.route('/history')
def history():
    return render_template('history.html', vals = urls.query.all())
```

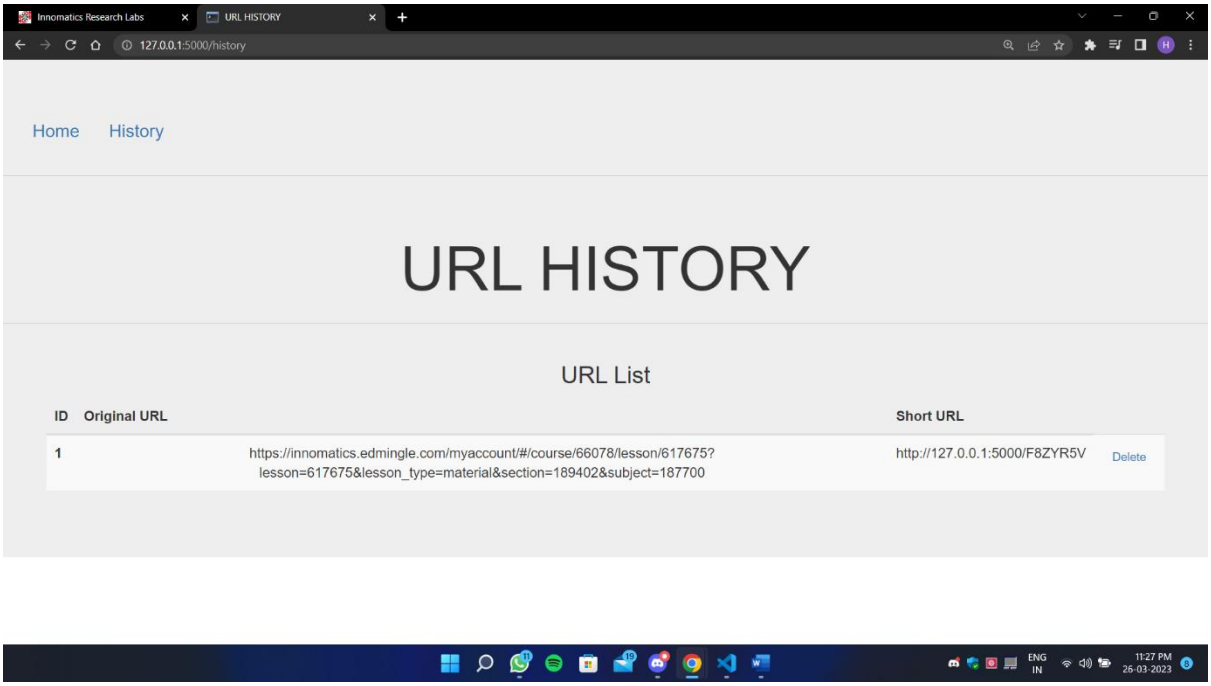
6. **Run The App**

```
# Run the App

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

Outputs:




Innomatics Research Labs
URL SHORTENER
127.0.0.1:5000/display/F8ZYR5V
Home
History
URL SHORTENER
Please Enter the URL
https://
Convert to Short URL
Short URL
http://127.0.0.1:5000/F8ZYR5V


Innomatics Research Labs
URL HISTORY
127.0.0.1:5000/history
Home
History
URL HISTORY
URL List
ID
Original URL
Short URL
1
https://innomatics.edmingle.com/myaccount/#/course/66078/lesson/617675?lesson=617675&lesson_type=material&section=189402&subject=187700
http://127.0.0.1:5000/F8ZYR5V
Delete