URL SHORTENING APPLICATION

PROJECT

Name:ALMAS BANU
Data Science Intern @Innomatics Research Labs.

URL Shortener Web Application For Basic Users

URLs. Project Workflow

1. Users can enter the URL they want to shorten. After entering a URL, click on the ‘Shorten’ URL button to display the shortened URL in the following text-field which can be copied by clicking on the copy button.
2. After the ‘Shorten’ button is clicked, the URL that is entered is saved in our database with the shortened URL. It is saved in the database so that the user can look into the previous URLs he entered in our web-app with their shortened URL.
3. Try to verify the URL entered by the user is correct or not. (Do some googling to find out how to make it possible)

FRONT-END INFORMATION.

The front-end consists of 2 web pages:

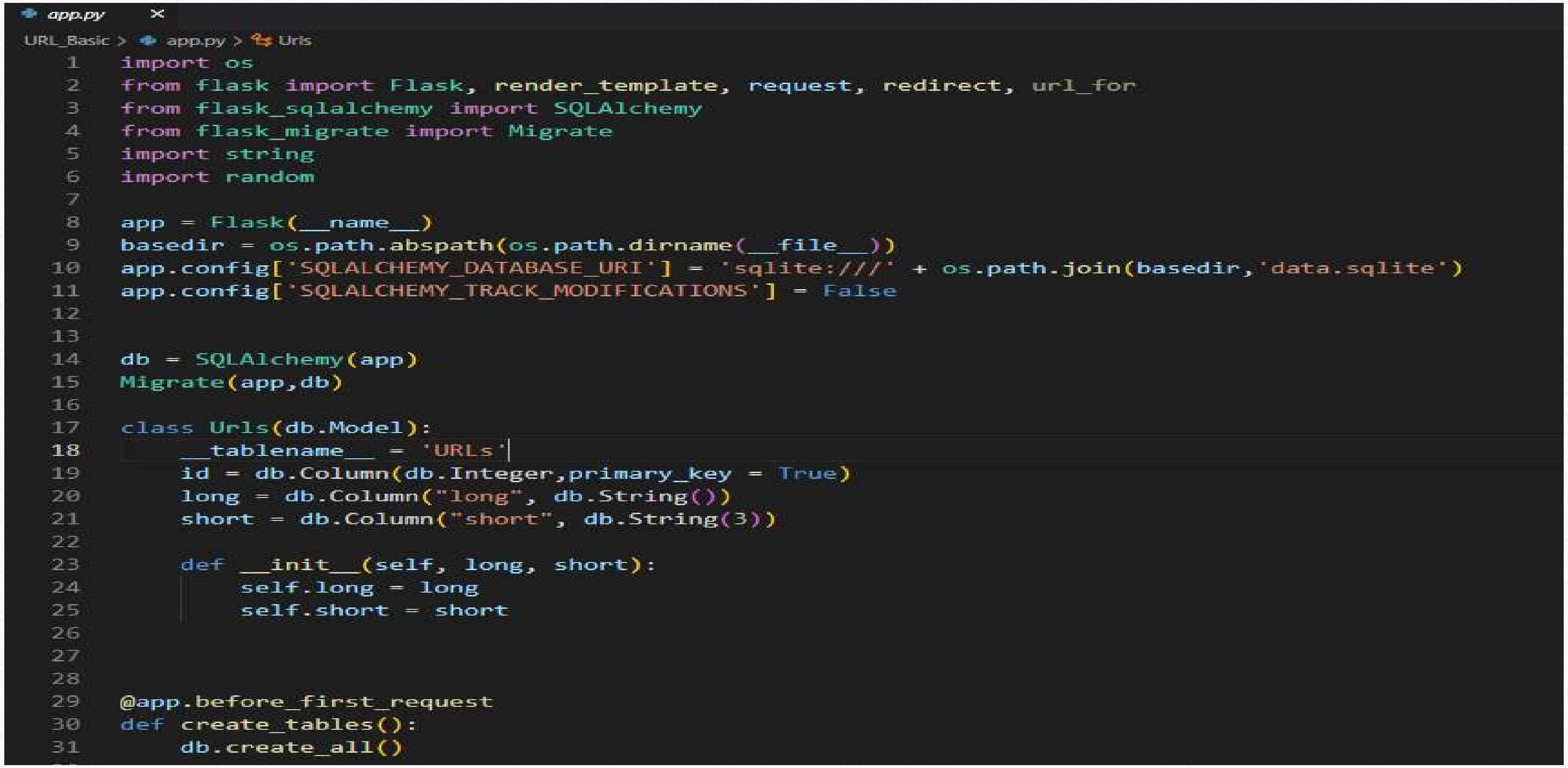
1. Home Page - A page will be shown where the user can enter the URL he/she wants to shorten. After the 'shorten' button is clicked, the shortened URL is displayed in the text-field which the user can copy using the copy button.
2. History Page - Containing all the Original URLs along with the Shortened URLs.

Programming Language used:Python,flask
Database : FLASK SQLITE

Modules used are:

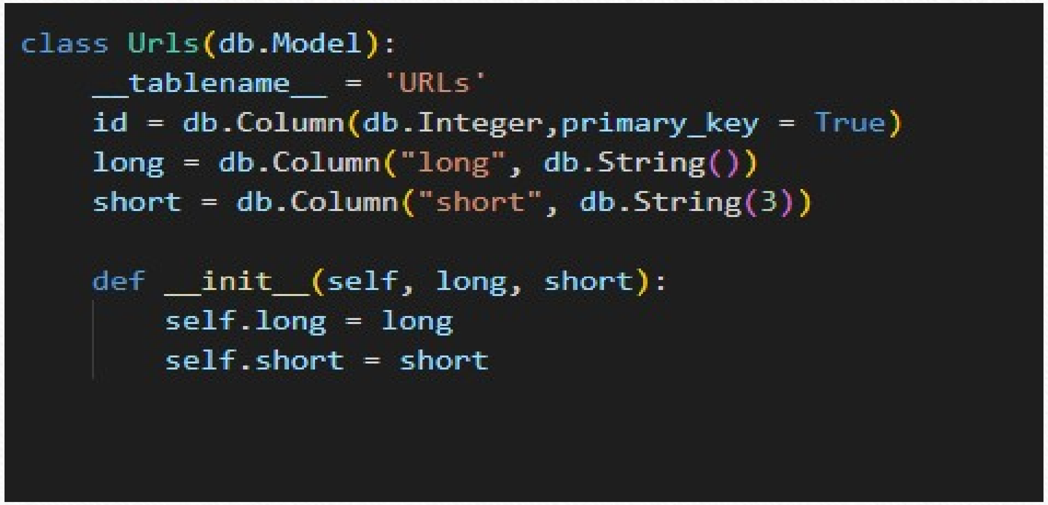
- os module
- from flask import Flask, render_template, request, redirect, url_for
- from flask_sqlalchemy import SQLAlchemy
- from flask_migrate import Migrate
- string and random module.

The main python file is app.py The application creation using flask,database creation(Urls),table creation,app configuration is done as below.



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

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


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

class Urls(db.Model):
    __tablename__ = 'URLs'
    id = db.Column(db.Integer,primary_key = True)
    long = db.Column("long", db.String())
    short = db.Column("short", db.String(3))

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



@app.before_first_request
def create_tables():
    db.create_all()
```

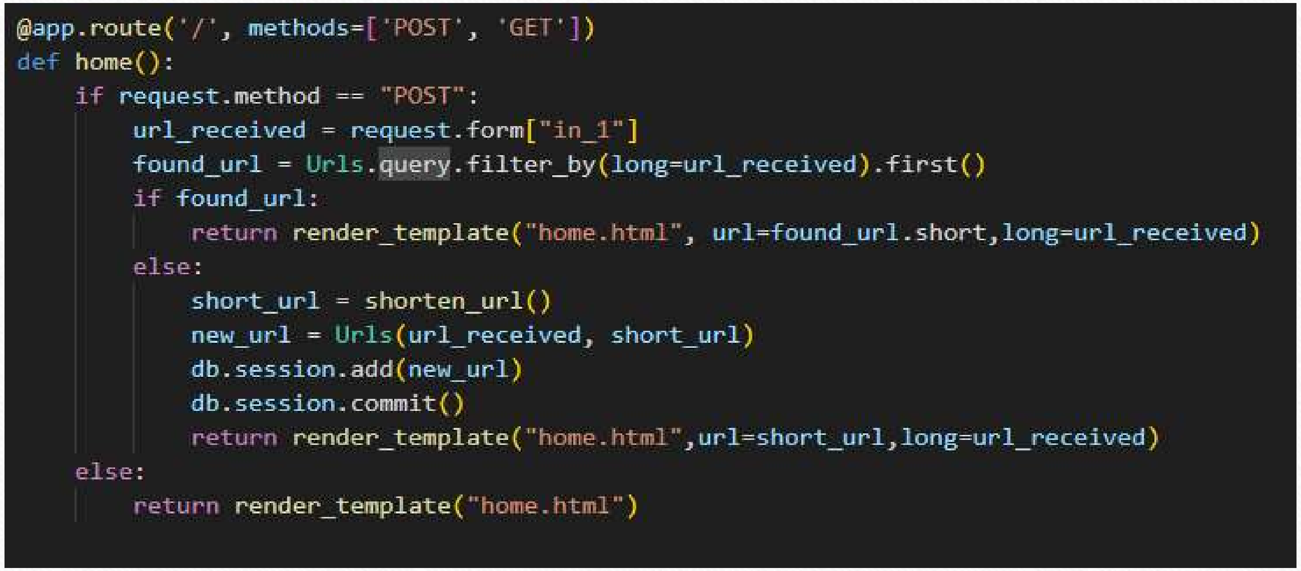
Class Urls where table name specified,id long for Original url and short for Shortened URL.



```
class Urls(db.Model):
    __tablename__ = 'URLs'
    id = db.Column(db.Integer,primary_key = True)
    long = db.Column("long", db.String())
    short = db.Column("short", db.String(3))

    def __init__(self, long, short):
        self.long = long
        self.short = short
```

Default route '/' is binded with home() function which receives the input saves in database and sents the input to shorten_url() function which converts long url to short and the short url is added to database and displayed on home.html page.If a user has already entered an long url whose short url is present in database then no conversion direct display of shorten url from database.

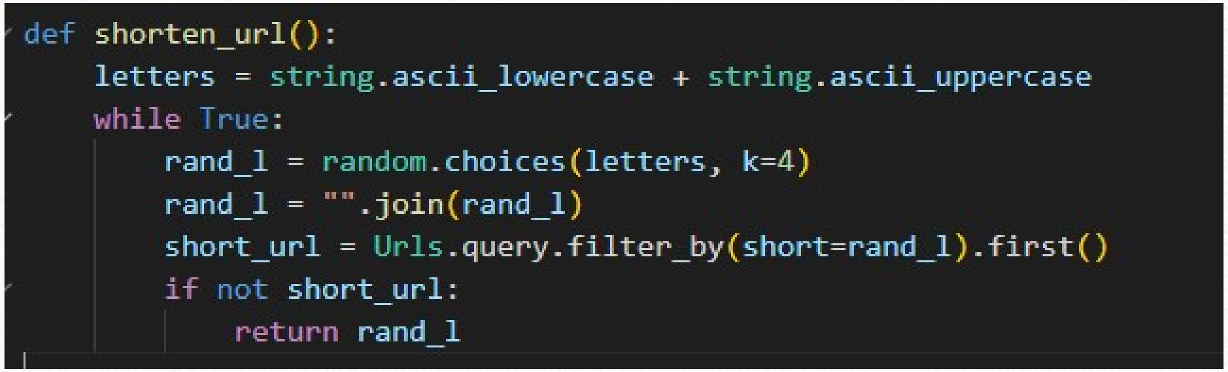
For default route GET method and POST method for taking input.



```python
@app.route('/', methods=['POST', 'GET'])
def home():
    if request.method == "POST":
        url_received = request.form["in_1"]
        found_url = Urls.query.filter_by(long=url_received).first()
        if found_url:
            return render_template("home.html", url=found_url.short,long=url_received)
        else:
            short_url = shorten_url()
            new_url = Urls(url_received, short_url)
            db.session.add(new_url)
            db.session.commit()
            return render_template("home.html",url=short_url,long=url_received)
    else:
        return render_template("home.html")
```

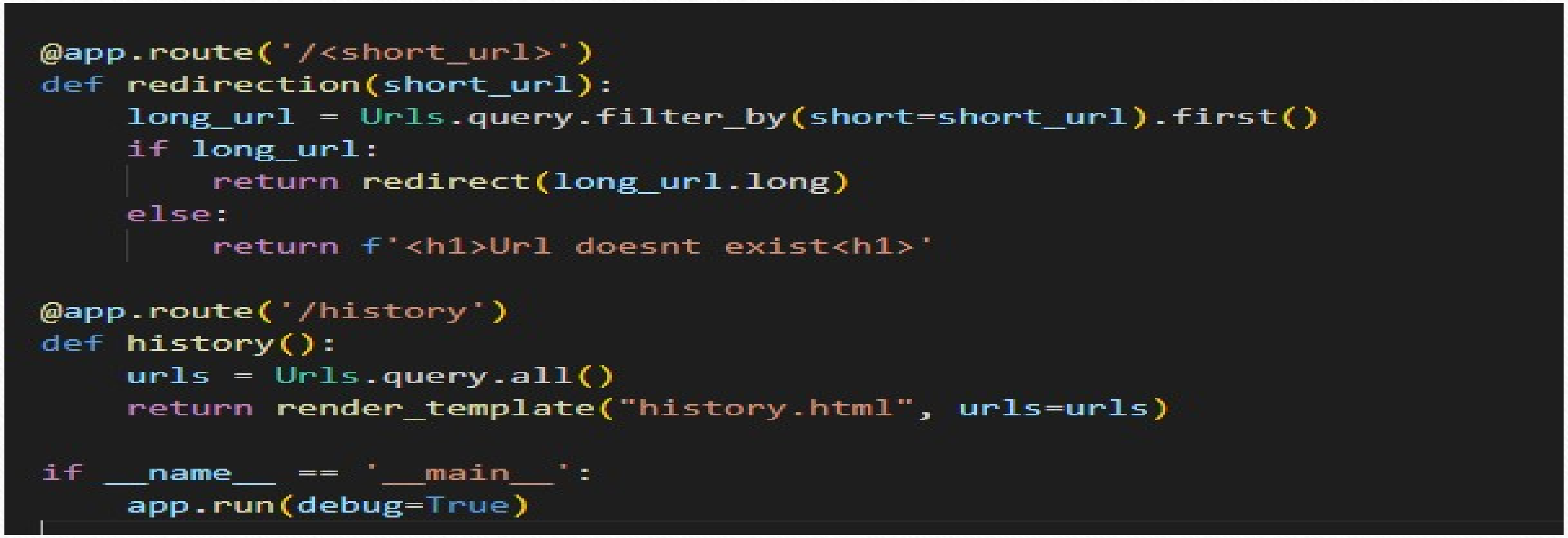
shorten_url() function:

In this function 4 random letters are generated for the long url using random function and returned as 4 letter code.When http://127.0.0.1:5000/4lettercode is clicked it redirects to long url page.



```
def shorten_url():
    letters = string.ascii_lowercase + string.ascii_uppercase
    while True:
        rand_l = random.choices(letters, k=4)
        rand_l = "".join(rand_l)
        short_url = Urls.query.filter_by(short=rand_l).first()
        if not short_url:
            return rand_l
```

Redirection function() for redirecting to original url page.
Used Urls.query_filter_by() function to search in database.
When short url is entered that has been redirected for Long URL PAGE.
/history route binded with history function which displays history.html that has list of Original URL’s and Shortened URL’s within the database.
And Finally running the app



```
@app.route('/<short_url>')
def redirection(short_url):
    long_url = Urls.query.filter_by(short=short_url).first()
    if long_url:
        return redirect(long_url.long)
    else:
        return f'<h1>Url doesnt exist<h1>'

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

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

About HTML Pages.

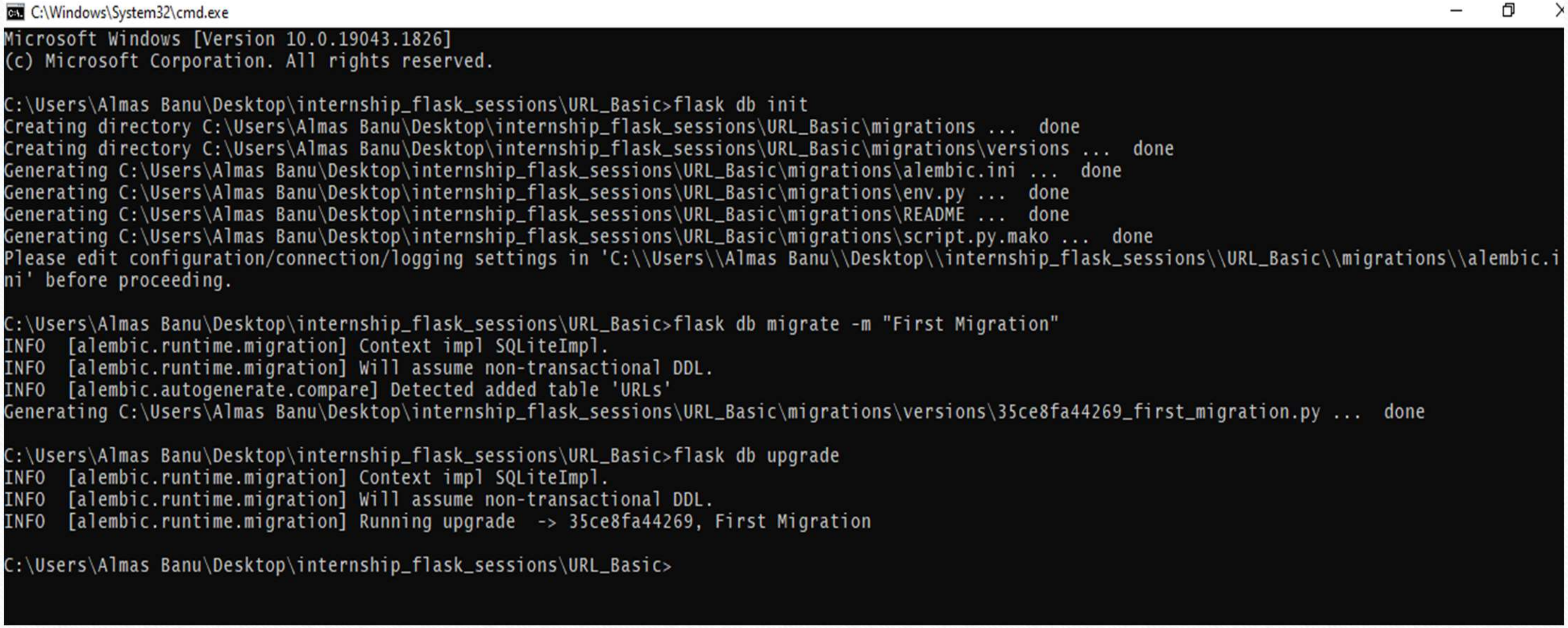
layout.html in templates folder from this file the remaining html files are inheriting the contents in order to obtain html files on website we have to use render_template function. Layout html has basic html,bootstrap required for website.It has block title,nav bars for Home and History page,block content.

home.html inherits from layout.html it takes input url from user when Shorten button is clicked along with original URL shortened URL is obtained.We have a Copy button to copy shortened URL.Copy button is created using javascript techniques.

history.html has a table with ID column,Original URL and Shortened URL created using css and html.
Shortened URL is combination of localhost IP + 4letterCode.

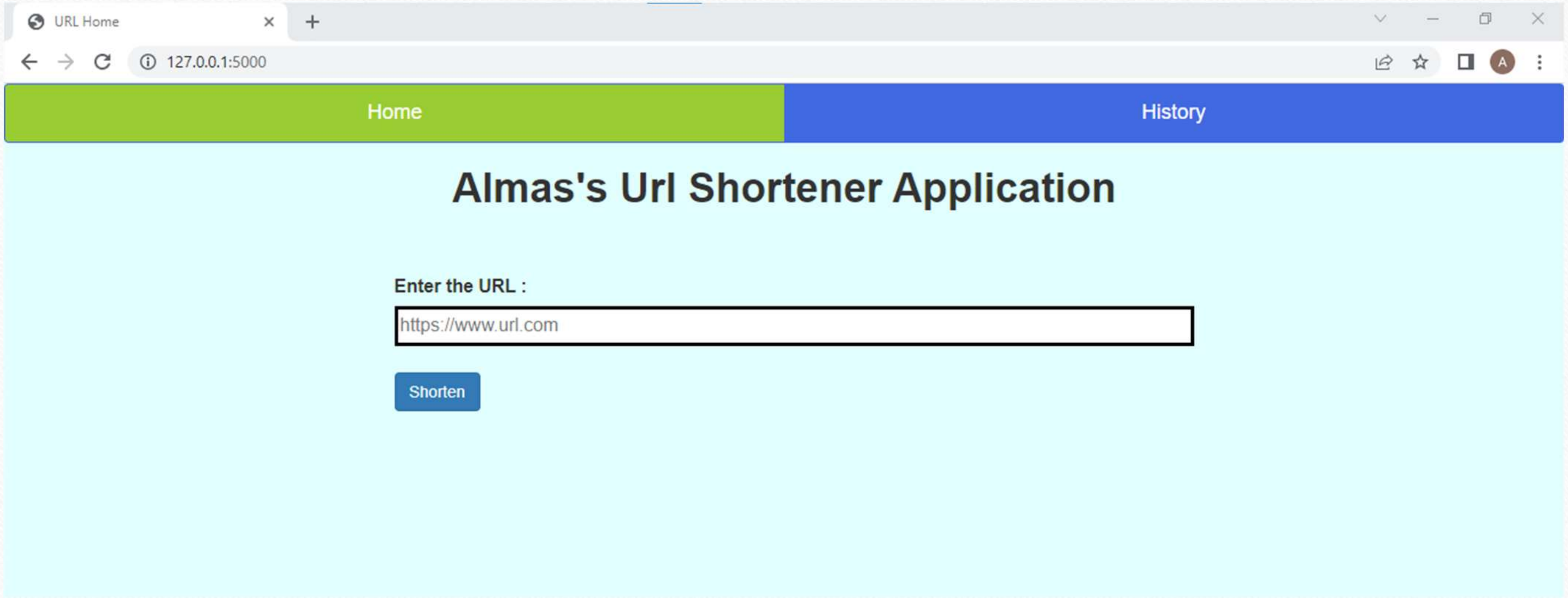
We can navigate to both Home and History Page.

Database creation using commands: flask db init,flask db migrate –m “Message” and flask db upgrade.



```
Microsoft Windows [Version 10.0.19043.1826]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Almas Banu\Desktop\internship_flask_sessions\URL_Basic>flask db init
Creating directory C:\Users\Almas Banu\Desktop\internship_flask_sessions\URL_Basic\migrations ...  done
Creating directory C:\Users\Almas Banu\Desktop\internship_flask_sessions\URL_Basic\migrations\versions ...  done
Generating C:\Users\Almas Banu\Desktop\internship_flask_sessions\URL_Basic\migrations\alembic.ini ...  done
Generating C:\Users\Almas Banu\Desktop\internship_flask_sessions\URL_Basic\migrations\env.py ...  done
Generating C:\Users\Almas Banu\Desktop\internship_flask_sessions\URL_Basic\migrations\README ...  done
Generating C:\Users\Almas Banu\Desktop\internship_flask_sessions\URL_Basic\migrations\script.py.mako ...  done
Please edit configuration/connection/logging settings in 'C:\\Users\\Almas Banu\\Desktop\\internship_flask_sessions\\URL_Basic\\migrations\\alembic.i
ni' before proceeding.

C:\Users\Almas Banu\Desktop\internship_flask_sessions\URL_Basic>flask db migrate -m "First Migration"
INFO  [alembic.runtime.migration] Context impl SQLiteImpl.
INFO  [alembic.runtime.migration] Will assume non-transactional DDL.
INFO  [alembic.autogenerate.compare] Detected added table 'URLs'
Generating C:\Users\Almas Banu\Desktop\internship_flask_sessions\URL_Basic\migrations\versions\35ce8fa44269_first_migration.py ...  done

C:\Users\Almas Banu\Desktop\internship_flask_sessions\URL_Basic>flask db upgrade
INFO  [alembic.runtime.migration] Context impl SQLiteImpl.
INFO  [alembic.runtime.migration] Will assume non-transactional DDL.
INFO  [alembic.runtime.migration] Running upgrade  -> 35ce8fa44269, First Migration

C:\Users\Almas Banu\Desktop\internship_flask_sessions\URL_Basic>
```

DEMO :



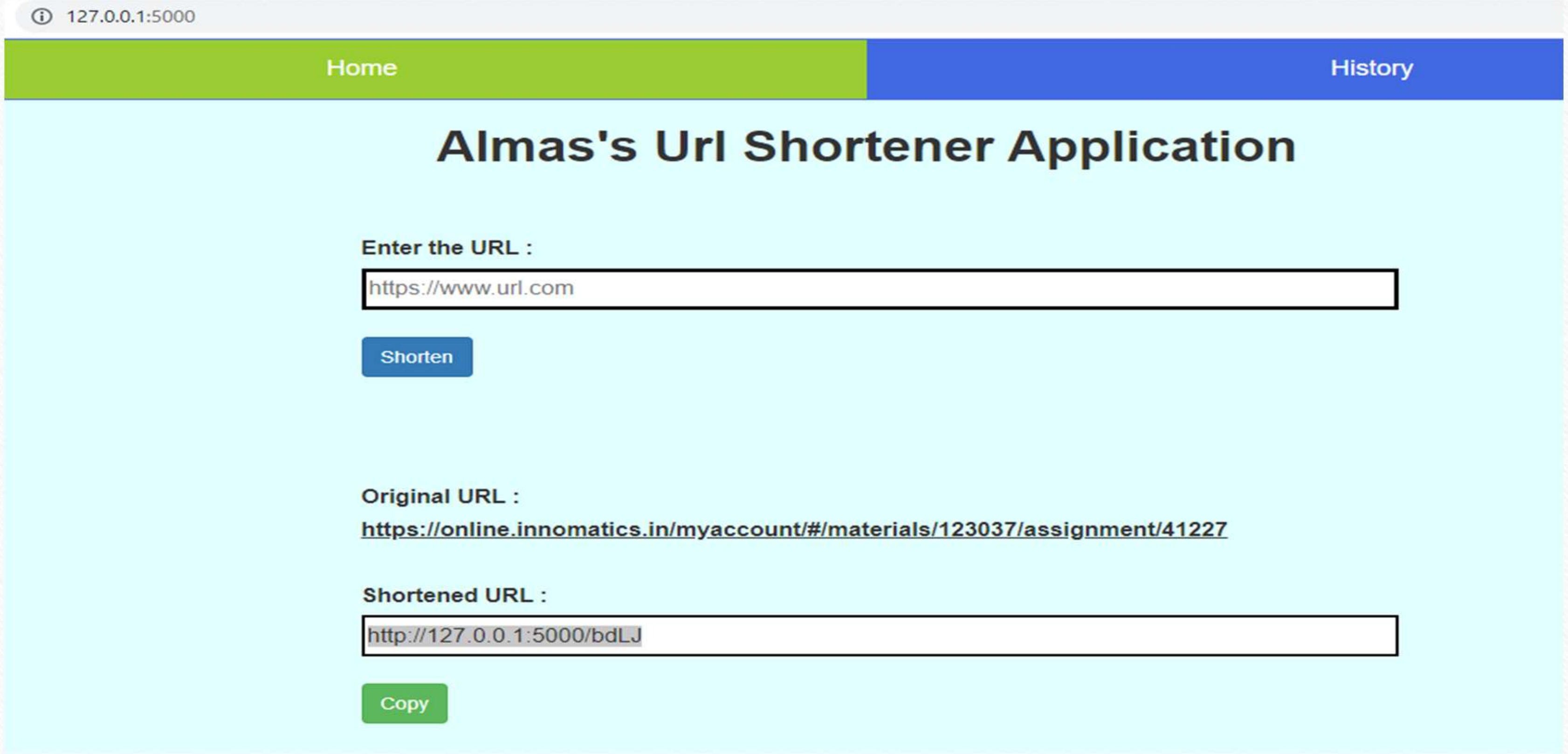
Almas's Url Shortener Application

Enter the URL :

https://www.url.com

Home:

Entered URL got Output and Copied Short URL



Almas's Url Shortener Application

Enter the URL :

https://www.url.com

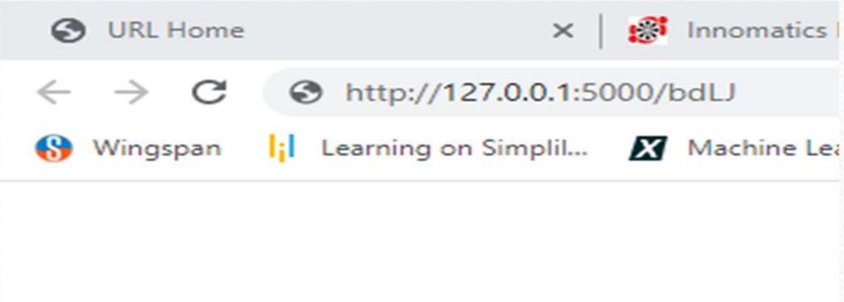
Original URL :

https://online.innomatics.in/myaccount/#/materials/123037/assignment/41227

Shortened URL :

http://127.0.0.1:5000/bdLJ

When pasted the copied URL redirected to original Long URL Page.



URL Home ×

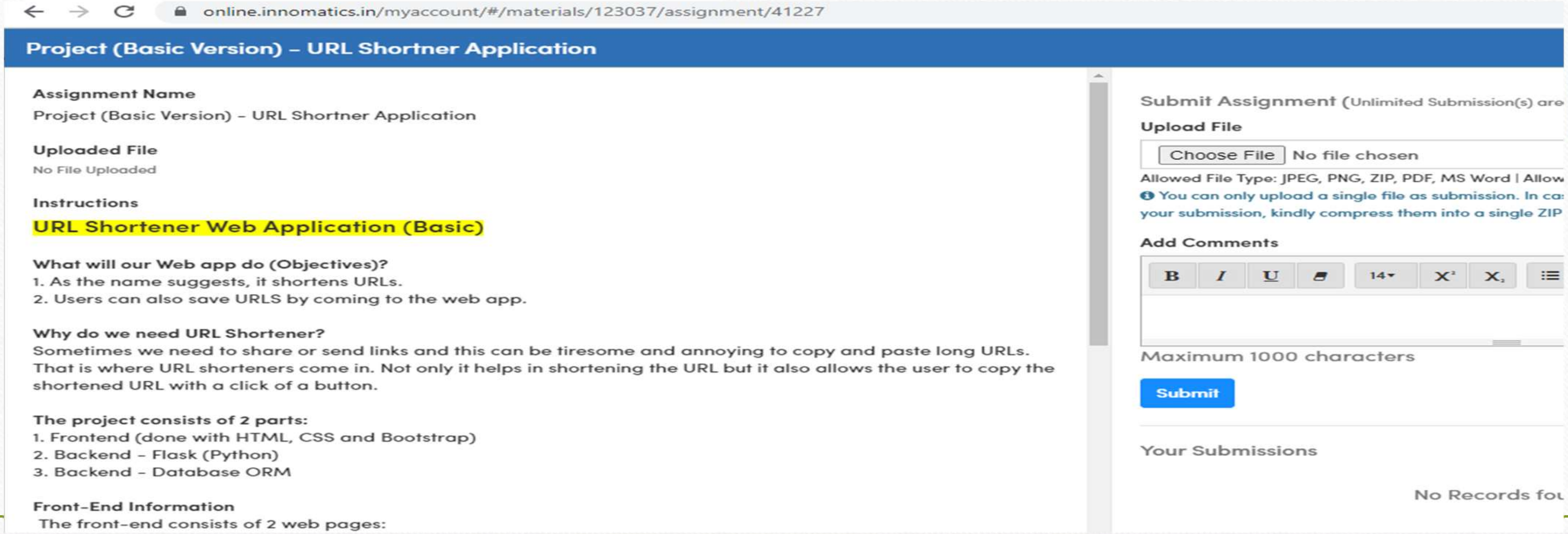
Innomatics

http://127.0.0.1:5000/bdLJ

Wingspan

Learning on Simplil...

Machine Lea



online.innomatics.in/myaccount/#/materials/123037/assignment/41227

Project (Basic Version) – URL Shortner Application

Assignment Name

Project (Basic Version) – URL Shortner Application

Uploaded File

No File Uploaded

Instructions

URL Shortener Web Application (Basic)

What will our Web app do (Objectives)?

1. As the name suggests, it shortens URLs.
2. Users can also save URLS by coming to the web app.

Why do we need 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 2 parts:

1. Frontend (done with HTML, CSS and Bootstrap)
2. Backend – Flask (Python)
3. Backend – Database ORM

Front-End Information

The front-end consists of 2 web pages:

Submit Assignment (Unlimited Submission(s) are

Upload File

Choose File No file chosen

Allowed File Type: JPEG, PNG, ZIP, PDF, MS Word | Allow

You can only upload a single file as submission. In ca

your submission, kindly compress them into a single ZIP

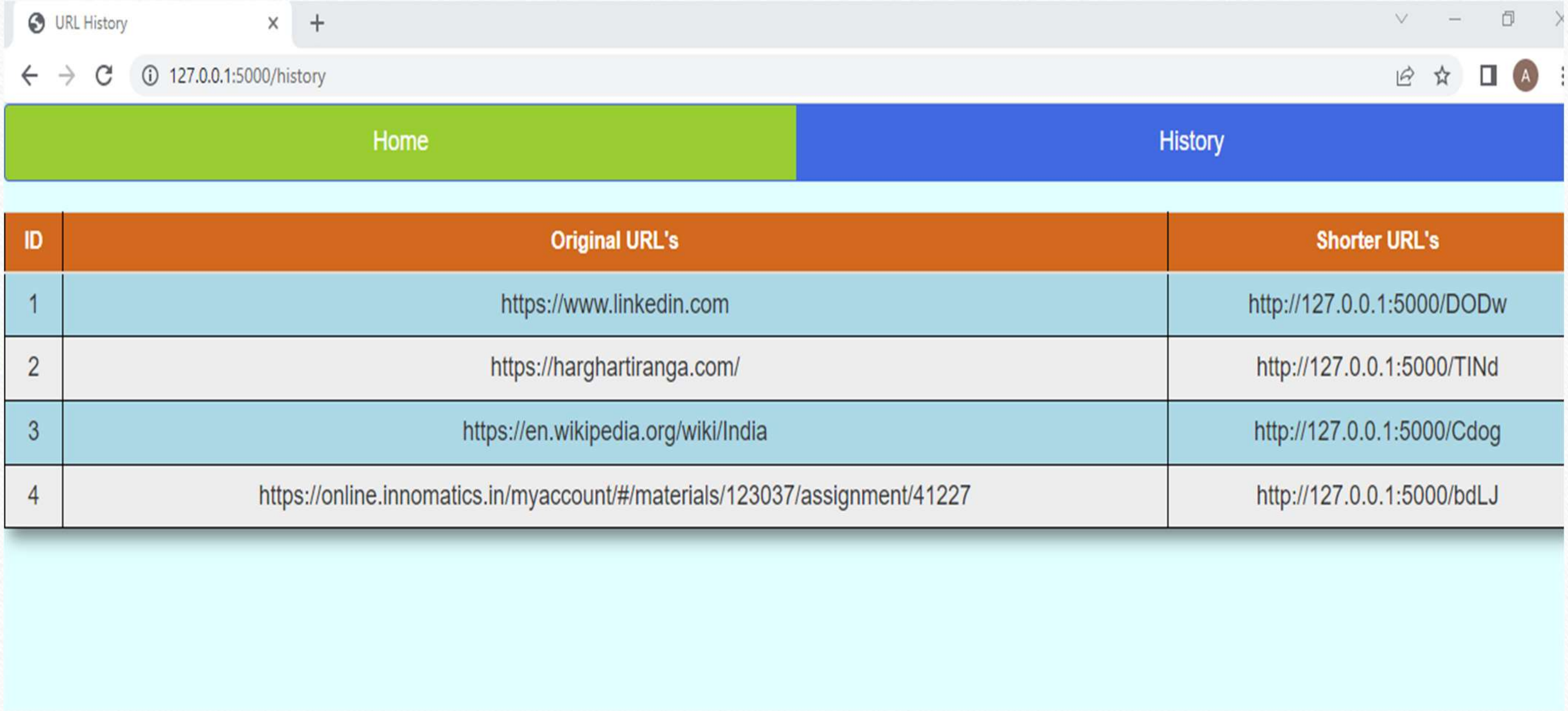
Add Comments

Maximum 1000 characters

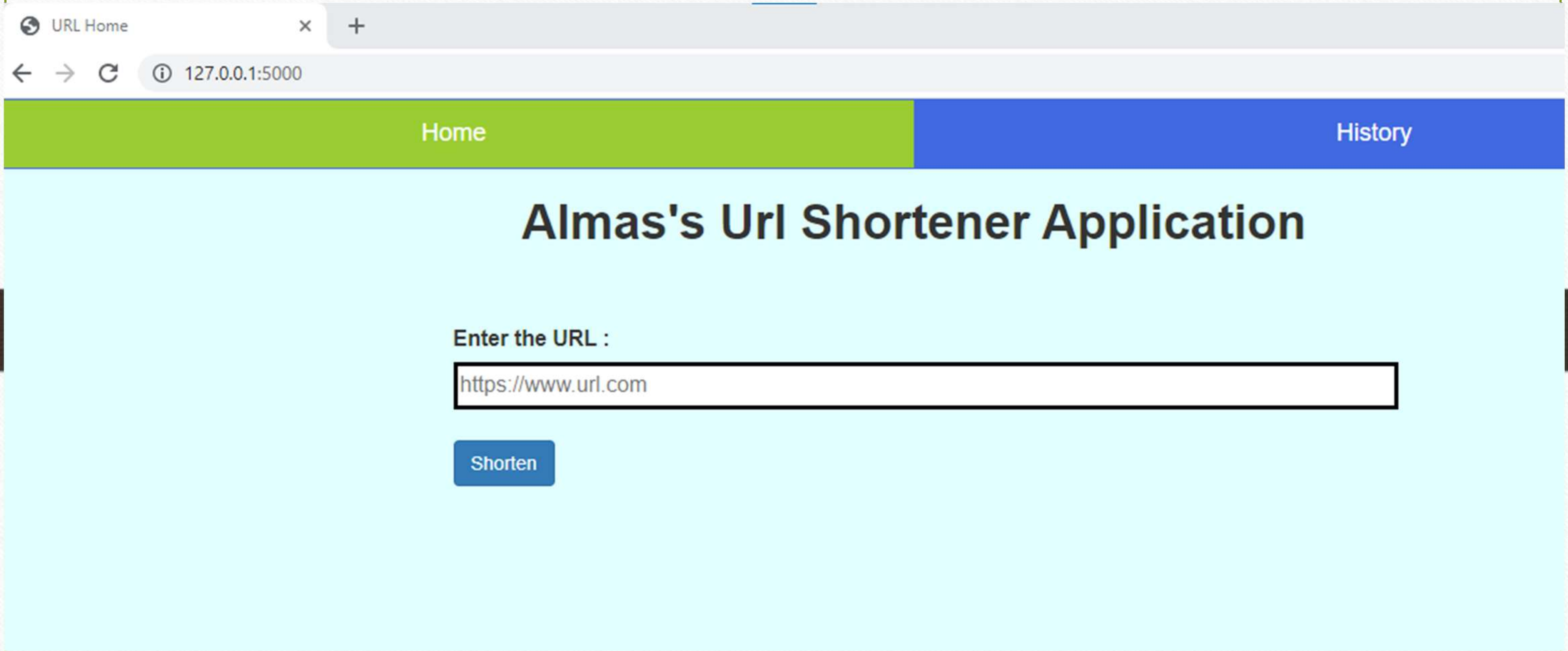
Your Submissions

No Records fou

History Page.



ID	Original URL's	Shorter URL's
1	https://www.linkedin.com	http://127.0.0.1:5000/DODw
2	https://harghartiranga.com/	http://127.0.0.1:5000/TINd
3	https://en.wikipedia.org/wiki/India	http://127.0.0.1:5000/Cdog
4	https://online.innomatics.in/myaccount/#/materials/123037/assignment/41227	http://127.0.0.1:5000/bdLJ



Almas's Url Shortener Application

Enter the URL :

https://www.url.com

THANK YOU

- Special thanks to
- My mentor – KANAV BANSAL
- Innomatics Research Labs.