

URL Shortener Web App Development

REPORT

CONTENT

- **About**
- **Task Information**
- **Approaches to solve the task**
- **Functionality or Features**

About:

I am Himanshu Vaish, Data Science Intern at Innomatics Research Labs under the leadership of Mr. Kanav Bansal sir. A task has been given by Mr. Kanav Bansal sir to develop a Web Application that will work as a URL shortener and to make a detailed report of approaches to solve the task and the challenges faced while completion of the task.

TASK INFORMATION:

To develop a web app that can shorten a URL provided by user and can store the history of the URL that are asked to shorten and the shorten URLs for every different user.

The project consists of 2 parts:

1. Frontend (done with HTML, CSS and Bootstrap)
2. Backend - Flask (Python)
 - Database - ORM (Object Relation Mapper)

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.

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).

Approaches to solve the Task:

To make this web app I have used Flask microframework integrated with ORM database. To style the app, I have used CSS and to give a schema to the app I have used HTML. A touch of Bootstrap and Jinja templating was also used to maintain the proper functionality of the app.

Steps I used to make this Web App:

- Flask app formation
- Database creation
- Database integration
- App's schema formation using HTML
- App's styling using CSS

I used Flask to make app that can run on my location machine for development purpose. I first of all created a virtual environment for this task so that I could not miss any required library or module to import. To create a database that can store the login info of user I used ORM (Object Relation Mapper) and then integrated it with the app.

I used HTML and CSS to make it available for user at browser end.

Functionality or Features of the App:

This app has following functionality:

- It has a login page that means if a user already have an account then he/she can login to dashboard
- It has a Register option, if a user is new and want to make a account then he/she can do it by clicking on to the “Register” button in the Navigation bar.
- If a user do not want to Register then also he/she can use the app to shorten the URL.
- By default user is get landed at app’s main page where he/she can directly use the app.
- A kind of history is also maintained for every user to let them track their original as well as shorten URLs

Here are some images of the app.





