**INNOMATICS RESEARCH LABS**

## INTERNSHIP PROJECT REPORT

**ON**

## WEB BASED URL SHORTENER

**SUBMITTED BY**

**MAGANTI HARISH**

**BATCH OCT – 21**

**GUIDED BY**

**KANAV BANSAL**

# Summary

**URL stands for uniform resource locater.** A URL shortener is a simple tool that takes a long URL and turns it into whatever URL you would like it to be. A custom, or branded URL shortener, is when you’ve connected your own custom domain to a URL shortener which acts as a base for all the short links you create. Instead of using a generic domain such as bit.ly or rebrand.ly, you can pick your own.

**Link tracking has got to be the number one reason to shorten a link**

# Purpose

There are several reasons to use URL shortening. Often regular un shortened link may be aesthetically unpleasing. Many web developers pass descriptive attribute in the URL to represent data hierarchy, command structure, transition path or session information. This can result in URL’s that are hundred of character long and contains complex character pattern. Such URL’s are difficult to memorize, type out or distribute.

# Advantages

Shortened URLs can offer a unique opportunity for your business. Several shortening services provide a custom domain brand. It means is that instead of your shortened URL appearing as an Ow.ly or Bit.ly link, you can create your URL.

# Goal

To Build the system which allows user to enter a URL.The URL should be valid URL.

# Technologies

1. Python
2. Flask
3. HTML 5
4. CSS
5. JavaScript
6. MySQL

# Tools

1. PyCharm IDE
2. Web Browser
3. Notepad

# Working Phases

1. URL input – The goal of the URL input is to accept the valid URL from a user and pass it to the URL shortening module to shorten the URL.
2. URL shortening -- The goal of this phase is to map the given URL with max of 10 unique alphanumeric character.
3. URL Mapping – To Map the short and Long URL in database or dictionary.
4. URL redirection – To Test whether the Shorten URL redirect to its address

or path correctly or not.

**Procedure And libraries :**

1.Import flask

2.create new flask app

3.initiate Data base

4.take input from the user (url)

5.if the url valid then look at that url present in the database ,if we kept earlier then display the shortened url to the user

6.if the url not there in the data base then map the original url in the database and store there ,then create domain name with that url and display to the user

7.if the entered url not valid then simply display url is invalid