

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfilment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering

By
R. SWETHA (40731110)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU

APRIL 2023



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI

CHENNAI- 600119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **R. Swetha (40731110)** who carried out the project entitled "**Full Stack Web App Development (TUFIDCO)**" under my supervision from March 2023 to April 2023.

Internal Guide

Ms. Hema. V, M.E.

Head of the Department

Dr. S. Vigneshwari, M.E., Ph.D.

Submitted for Viva voce Examination held on 17-04-2023

Internal Examiner

External Examiner

DECLARATION

I Swetha. R hereby declare that the Project Report entitled 'Full Stack Web App Development' done by me under the guidance of **Ms. Hema. V, M.E.**, at '**TUFIDCO – Tamil Nadu Urban Finance and Infrastructure Development Corporation Limited**', is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE: 19/04/2023



PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sashikala M.E., Ph.D., Dean, School of Computing, Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. Hema. V, M.E.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

INTERNSHIP CERTIFICATE

Tamilnadu Urban Finance and Infrastructure Development Corporation Limited
CIN: U65191TN1990SGC018909
(A Government of Tamilnadu Undertaking)
Regd. Office : No. 490/1-2, Anna Salai, Nandanam, Chennai - 600 035.

Phone : 24329800, 24329801
24329802
Fax : 24350814
Grams : TUFIDCO
E-mail : tufidco@tn.gov.in

Lr.No.TUFIDCO/SSM(M)/2023

Date: 20-04-2023

To
Dr.L.Lakshmanan,
Professor and Head of Department,
Department of Computer Science and Engineering,
School of Computing,
Sathyabama Institute of Science and Technology, Chennai - 600 035.

Sir,

Sub: Internship for the students of Sathyabama Institute of Science and Technology, Chennai – Reg.
Ref: SHD/2023/01/20/A9, 20-01-2023

With reference to your letter cited, it is hereby informed that, the following students of the Department of Computer Science and Engineering, School of Computing, Sathyabama Institute of Science and Technology, Chennai are hereby permitted to undergo Internship at TUFIDCO for a period from 6th March 2023 to 6th April 2023.

Sl. No.	Name of the Student	Reg. No.	Department	Year
1	Sriram.R	40731103	BE CSE	Third
2	Sukesh Raghav.G	40731106	BE CSE	Third
3	Swetha.R	40731110	BE CSE	Third
4	Tejaswini.T	40731112	BE CSE	Third

The interns have worked on TUFIDCO website development under the supervision of Mrs.M.Malarvizhi, Senior System Manager. During their internship, they gained considerable knowledge based on Website design and development.

We hereby certify their overall work as excellent to the best of our knowledge and wishing them the best luck in their future endeavors.

Yours faithfully,
For TUFIDCO


General Manager

tufidco
Web : www.tufidco.in

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	1
	LIST OF FIGURES	2
	LIST OF ABBREVIATIONS	3
1	INTRODUCTION	4
	1.1 WHAT IS FLASK?	4
	1.2 REQUIREMENTS FOR FLASK PROJECT.	7
	1.3 FULL STACK WEB APP DEVELOPMENT(TUFIDCO)	11
	1.4 OVERVIEW OF PROJECT	11
2	AIM AND SCOPE OF THE PRESENT INVESTIGATION	17
	2.1 AIM	17
	2.2 SCOPE	17
	2.3 OBJECTIVE	17
3	EXPERIMENTAL OR MATERIALS AND METHODS	18
	3.1 HARDWARE REQUIREMENTS	18
	3.2 SOFTWARE REQUIREMENTS	18
	3.3 MODULE DESCRIPTION	18
	3.4 METHODOLOGY	19
	3.5 FUNCTIONAL REQUIREMENTS	23
4	SUMMARY AND CONCLUSION	24
	APPENDIX	
	A. OVERVIEW OF COMPONENTS AND SCREENSHOTS	25
	B. SOURCE CODE	31

ABSTRACT

This abstract describes a government website that has been built using Flask, Bootstrap, HTML, CSS, JavaScript, and a SQL database. The website is designed to provide a range of services to citizens, including access to TUFIDCO services, information about TUFIDCO policies and programs, and news and updates about TUFIDCO activities.

The website's front-end has been developed using Bootstrap, HTML, CSS, and JavaScript. The website's design is responsive and user-friendly, ensuring that citizens can easily navigate the site and find the information they need. The front-end includes a range of different features, including search functionality, online forms, and multimedia content.

The website's back-end has been developed using Flask, a Python-based web framework. Flask provides a range of different tools and libraries that make it easy to build web applications quickly and efficiently. We have used Flask to build a range of different components, including a RESTful API, authentication and authorization features, and data processing tools.

The website's database has been built using SQL. This has enabled us to store and manage large amounts of data efficiently and securely. We have used mysql-connector to interact with MySQL database.

The TUFIDCO website we have developed is a powerful and flexible tool that can be used to provide citizens with the information and services they need. It provides a range of different features and is built using a range of powerful and flexible tools, making it easy to extend and customize as needed. Overall, this government website is a valuable resource for citizens and an essential tool for the government in providing effective and efficient public services.

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
4.1	Login page	46
4.2	Admin Dashboard	47
4.3	Add and Manage Student	49
4.4	View Attendance	53
4.5	Staff Dashboard	55
4.6	Student Dashboard	59
4.7	View Result	59
4.8	SQLite Connection	63

LIST OF ABBREVIATIONS

ABBREVIATION

SQL
DB
App
HTML
CSS
HTTP
URL
py

EXPANSION

Structures Query Language
Database
Application
Hypertext Markup Language
Cascading Style Sheets
Hypertext Transfer Protocol
Uniform Resource Locator
Python

CHAPTER 1: INTRODUCTION

1.1 WHAT IS A WEB FRAMEWORK?

A Web Application Framework or a simply a Web Framework represents a collection of libraries and modules that enable web application developers to write applications without worrying about low-level details such as protocol, thread management, and so on.

WHAT IS FLASK?

Flask is a micro web framework written in Python. It is designed to be lightweight and modular, with minimal dependencies. Flask is often used for building small to medium-sized web applications, APIs, and prototypes. It provides basic features such as routing, request and response handling, and template rendering, but also allows for easy extension through its many available plugins. Flask is known for its simplicity, flexibility, and ease of use, making it a popular choice for beginners and experienced developers alike.

WSGI

The Web Server Gateway Interface (Web Server Gateway Interface, WSGI) has been used as a standard for Python web application development. WSGI is the specification of a common interface between web servers and web applications.

jinja2

jinja2 is a popular template engine for Python. A web template system combines a template with a specific data source to render a dynamic web page.

This allows you to pass Python variables into HTML templates like this:

```
<html>
  <head>
    <title>{{ title }}</title>
```

```
</head>
<body>
  <h1>Hello {{ username }}</h1>
</body>
</html>
```

Microframework

Flask is often referred to as a microframework. It is designed to keep the core of the application simple and scalable.

Instead of an abstraction layer for database support, Flask supports extensions to add such capabilities to the application.

Unlike the Flask framework, Flask is very Pythonic. It's easy to get started with Flask, because it doesn't have a huge learning curve.

On top of that it's very explicit, which increases readability. To create the "Hello World" app, you only need a few lines of code.

This is a boilerplate code example.

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello World!'

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

If you want to develop on your local computer, you can do so easily. Save this program as server.py and run it with python server.py.

```
$ python server.py
* Serving Flask app "hello"
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

It then starts a web server which is available only on your computer. In a web browser open localhost on port 5000 (the url) and you'll see "Hello World" show up.

Micro means that the Flask framework is simple but extensible. You may all the decisions: which database to use, do you want an ORM etc.

Flask is one of the most popular web frameworks, meaning it's up-to-date and modern. You can easily extend its functionality. You can scale it up for complex applications.

1.2 REQUIREMENTS FOR FLASK PROJECT.

Prepare Your Environment

When you're ready to start your new Flask web application, create a new folder and navigate into it. In this folder, you'll set up a new virtual environment using your command line:

```
$ python3 -m venv env
```

This command sets up a new virtual environment named `env` in your current working directory. Once the process is complete, you also need to activate the virtual environment.

```
$ source env/bin/activate
```

If the activation was successful, then you'll see the name of your virtual environment, (`env`), at the beginning of your command prompt. This means that your environment setup is complete.

You can learn more about how to work with virtual environments in Python, and how to perfect your Python development setup, but for your Flask setup, you have all you need. You can continue with installing the flask package.

Install Flask and Pin Your Dependencies

Once you've created and activated your Python virtual environment, you can install Flask into this dedicated development workspace:

```
(env) $ python -m pip install flask
```

This command fetches the flask package from the Python Package Index (PyPI) using `pip`. After the installation has completed, you can pin your dependencies to make sure that you're keeping track of which Flask version you installed:

(env) \$ python -m pip freeze > requirements.txt

This command writes the names and versions of all external Python packages that are currently in your virtual environment to a file called requirements.txt. This file will include the flask package and all of its dependencies.

Note: There are many different versions of Flask. While it's usually best to work with the most recent version when starting a new project, you might have to work with a specific version for one particular project. You can install any version of Flask by adding the version number to the installation command:

(env) \$ python -m pip install flask==2.2.11

This command installs the version 2.2.11 of Flask to your environment instead of fetching the most recent version. Replace the number after the double equals sign (==) with the specific Flask version you need to install.

You should always include a record of the versions of all packages you used in your project code, such as in a requirements.txt file. The requirements.txt file allows you and other programmers to reproduce the exact conditions of your project build.

Suppose you're working on an existing project with its dependencies already pinned in a requirements.txt file. In that case, you can install the right Flask version as well as all the other necessary packages in a single command:

(env) \$ python -m pip install -r requirements.txt

The command reads all names and versions of the pinned packages from your requirements.txt file and installs the specified version of each package in your virtual environment.

Keeping a separate virtual environment for every project allows you to work with different versions of Flask for different web application projects. Pinning the dependencies with pip freeze enables you to reproduce the environment that you need for the project to work as expected.

Set Up a Flask Project

After you've successfully installed Flask, you're ready to create the scaffolding for your new web application. The Flask framework distinguishes between projects and apps.

A Flask project is a high-level unit of organization that contains logic that governs your whole web application. Each project can contain multiple apps.

A Flask app is a lower-level unit of your web application. You can have zero to many apps in a project, and you'll usually have at least one app. You'll learn more about apps in the next section.

With your virtual environment set up and activated and Flask installed, you can now create a project:

File Structure

```
my_project/
├── app/
│   ├── static/
│   └── templates/
├── .env
└── app.py
```

In the code block above, you can see the folder structure.

`app/` is your top-level project folder.

`app/app/` is your lower-level folder that represents your management app.

Inside the directory, there are several files:

- `app.py`: Initializes the Flask application instance and any extensions or modules used by the application and routes of the application
- `static/`: Contains static files such as CSS, JavaScript, and images.
- `templates/`: Contains Jinja2 templates used to generate HTML pages.
- `requirements.txt`: Contains the dependencies required by the project.

Start a Flask App

Every project you build with Flask can contain multiple Flask apps. When you ran the `startproject` command in the previous section, you created a management app that you'll need for every default project that you'll build. Now, you'll create a Flask app that'll contain the specific functionality of your web application.

(env) \$ python app.py

This command generates the web-page using localhost. Remember to use wsgi server for production.

1.3 FULL STACK WEB APP DEVELOPMENT (TUFIDCO)

The website developed using Flask acts as an information center that provides citizens with the information regarding various financial schemes offered by TUFIDCO- Tamil Nadu Urban Finance and Infrastructure Development Corporation Limited.

The Full Stack Web Application contains various jinja2 template codes that are used to display the information regarding the financial schemes.

The data is retrieved from a MySQL database.

1.4 OVERVIEW OF PROJECT

The Student Infotainment System Project in Django is developed using Python, Flask, HTML, CSS, Javascript and Bootstrap. This Project with source code is a complete and runnable system that contains different user interfaces. This system also has a good template design.

This project aims to display the information about the financial schemes. There are many components like about us, home, contact us, schemes, etc.