# Flask

# Ex 11

**2000**

Create branches and get the parts of the project in to the GIT, then send the Link of the your team repo.

# Build a Smart Website for Cancer Recognition Using Flask and Machine Learning

## Objective:

**The team will develop a web application using Flask that can recognize cancer based on user input. The website will include a home page, login, register, input, and result pages. Additionally, the team will train a machine learning model to perform cancer diagnosis based on a provided dataset.**

# Suggested Team Structure and Task Breakdown

**This project is split into five parts, with each team member taking responsibility for one part.**

# Part 1: Flask Project Setup and Home Page (Team Member 1)

### Task Title: Setup Flask Application and Create Home Page

## Responsibilities:

- Set up the Flask project structure (app.py, templates, static folders).
- Create a home page (/) that introduces the application and provides navigation links to the login, registration, and input pages.
- Ensure proper routing and basic structure of the Flask application.

# Part 2: User Registration and Login System (Team Member 2)

### Task Title: Implement User Registration and Login Using Flask-SQLAlchemy

## Responsibilities:

- Create the registration page (/register) and login page (/login).
- Implement user authentication, allowing users to create an account and log in using SQLite as the database.
- Ensure that access to the input page and prediction results is restricted to logged-in users only.
- Implement session management to track logged-in users.

# Part 3: Machine Learning Model Training and Integration (Team Member 3)

## Task Title: Train the Machine Learning Model and Integrate It into the Application

### Responsibilities:

- Train a machine learning model for cancer diagnosis using a provided dataset (e.g., breast cancer dataset from Scikit-learn).
- Save the trained model to a file (model.pkl) and load it into the Flask application.
- Create a route and functionality for the prediction page where users input features related to cancer diagnosis.
- Pass user input to the model and display the diagnosis (e.g., cancerous or non-cancerous) on the results page.

# Part 4: Input and Result Pages (Team Member 4)

## Task Title: Create and Handle Input and Result Pages

### Responsibilities:

- Design the input page (/input) where users provide information related to cancer diagnosis (e.g., age, tumor size, cell type, etc.).
- Validate the input and ensure it is passed to the machine learning model for prediction.
- Create a result page (/result) that displays the prediction provided by the machine learning model, along with an explanation of the diagnosis.

# Part 5: Database for Storing User Inputs and Predictions (Team Member 5)

## Task Title: Store User Inputs and Predictions in a SQLite Database

### Responsibilities:

- Create a database to store user inputs and the results of the cancer diagnosis.
- Link each prediction to the user who made the submission.
- Create a page (/history) where logged-in users can view their previous predictions and input data.
- Ensure that sensitive information (e.g., health data) is handled securely.