Exp No: 9	**Model Deployment: REST API with Flask and Containerization with Docker**

Aim:
To demonstrate the process of deploying a pre-trained machine learning model as a RESTful API using Flask, and then containerizing this API and model using Docker for easy deployment and portability.

Algorithm:
1. RESTful API Design (Flask)
Representational State Transfer (REST) is an architectural style for networked applications. A RESTful API uses standard HTTP methods (GET, POST, PUT, DELETE) to perform operations on resources.
Key Concepts:
Resources: Any data object that can be identified, named, addressed, or handled in the web. In our case, the ML model's prediction endpoint will be a resource.
Endpoints: Specific URLs that represent the resources.
HTTP Methods:
  `POST`: Used to submit data to a specified resource (e.g., send new data for prediction).
  `GET`: Used to request data from a specified resource (e.g., check API status).
Statelessness: Each request from a client to a server must contain all the information needed to understand the request. The server should not store any client context between requests.
JSON: JavaScript Object Notation is commonly used for data exchange between the client and the API.

2. Containerization with Docker
Docker is a platform that uses OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries, and configuration files.

Key Concepts:
Dockerfile: A text file that contains all the commands a user could call on the command line to assemble an image. It defines the environment, dependencies, and execution command for your application.
Image: A lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries, and settings.
Container: A runnable instance of an image. You can create, start, stop, move, or delete a container.
Port Mapping: Connecting a port on the host machine to a port inside the Docker container.

CODE:
OUTPUT: