Q1 #Flask is a lightweight web framework written in Python.

#It is designed to be simple and easy to use, yet flexible enough to handle a wide range of web applications.

#Flask follows the WSGI (Web Server Gateway Interface) standard and can run on any WSGI-compliant web server.

#Advantages of Flask Framework:

#Lightweight and Minimalistic: Flask has a small and simple core, making it easy to understand and use.

#It does not impose any particular way of doing things, allowing developers to have more flexibility in building their applications.

#Easy to Get Started: Flask provides a straightforward and intuitive API, making it easy for beginners to learn and start building web applications.

#The documentation is extensive and well-maintained, providing clear examples and tutorials.

#Modular and Extensible: Flask follows the "micro" philosophy, which means it provides only the essentials for building web applications.

#However, Flask is highly extensible, and developers can add additional functionality using Flask extensions or by integrating other Python libraries.

#Flexible Routing: Flask provides a powerful routing system that allows developers to define URL routes and map them to Python functions.

#This makes it easy to handle different URLs and HTTP methods and build RESTful APIs.

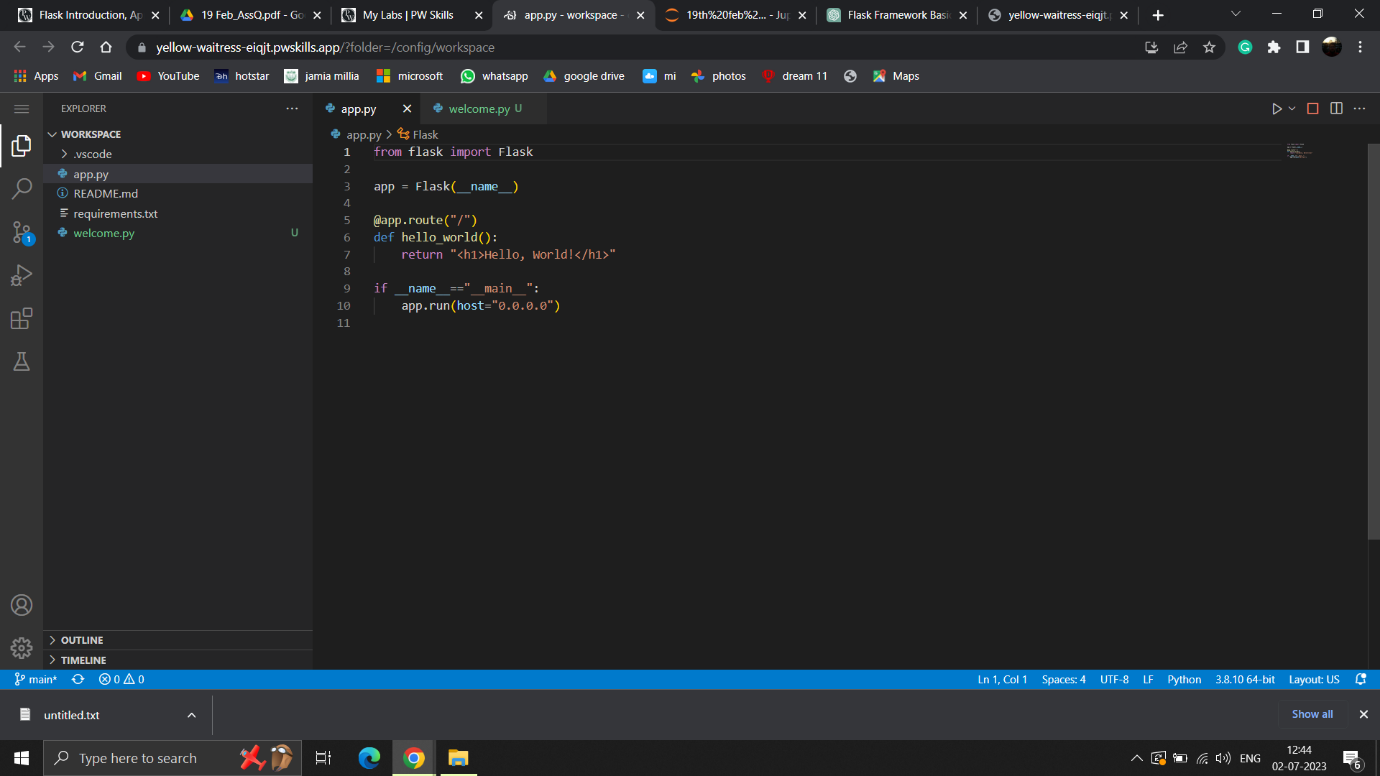
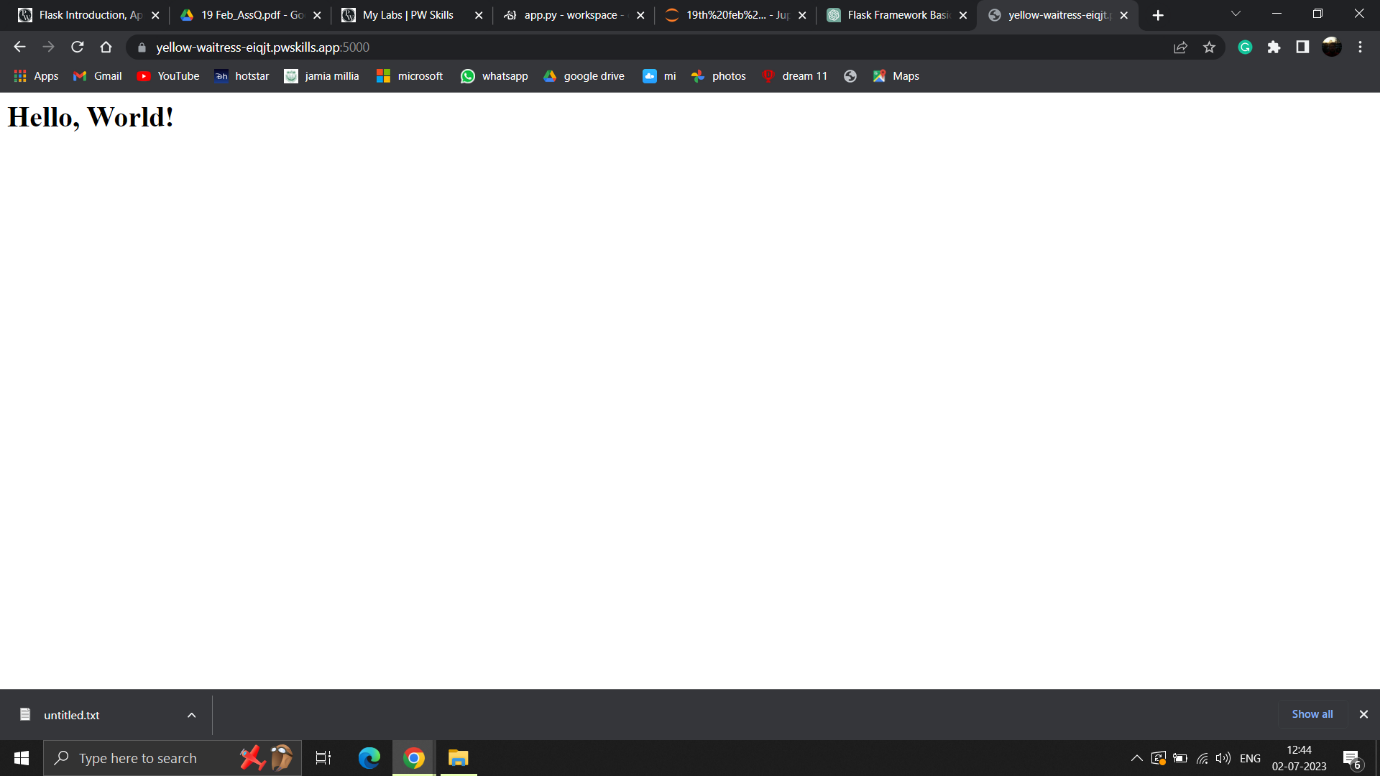
#Template Engine: Flask includes a template engine called Jinja2,

#which enables developers to separate the presentation logic from the business logic.

#It provides template inheritance, macros, and filters, making it easy to generate HTML and other dynamic content.

#Large Ecosystem: Despite being a lightweight framework, Flask has a large and active community.

#There are numerous third-party extensions available for adding features like database integration, authentication, and more

Q2

Q3 #In Flask, app routing refers to the process of mapping URL routes to specific functions in the application.

#It allows defining different routes for handling different URLs and HTTP methods (GET, POST, etc.).

#The app routes are defined using the @app.route() decorator in Flask.

#We use app routes in Flask to define the behaviour of the application based on different URLs and HTTP methods.

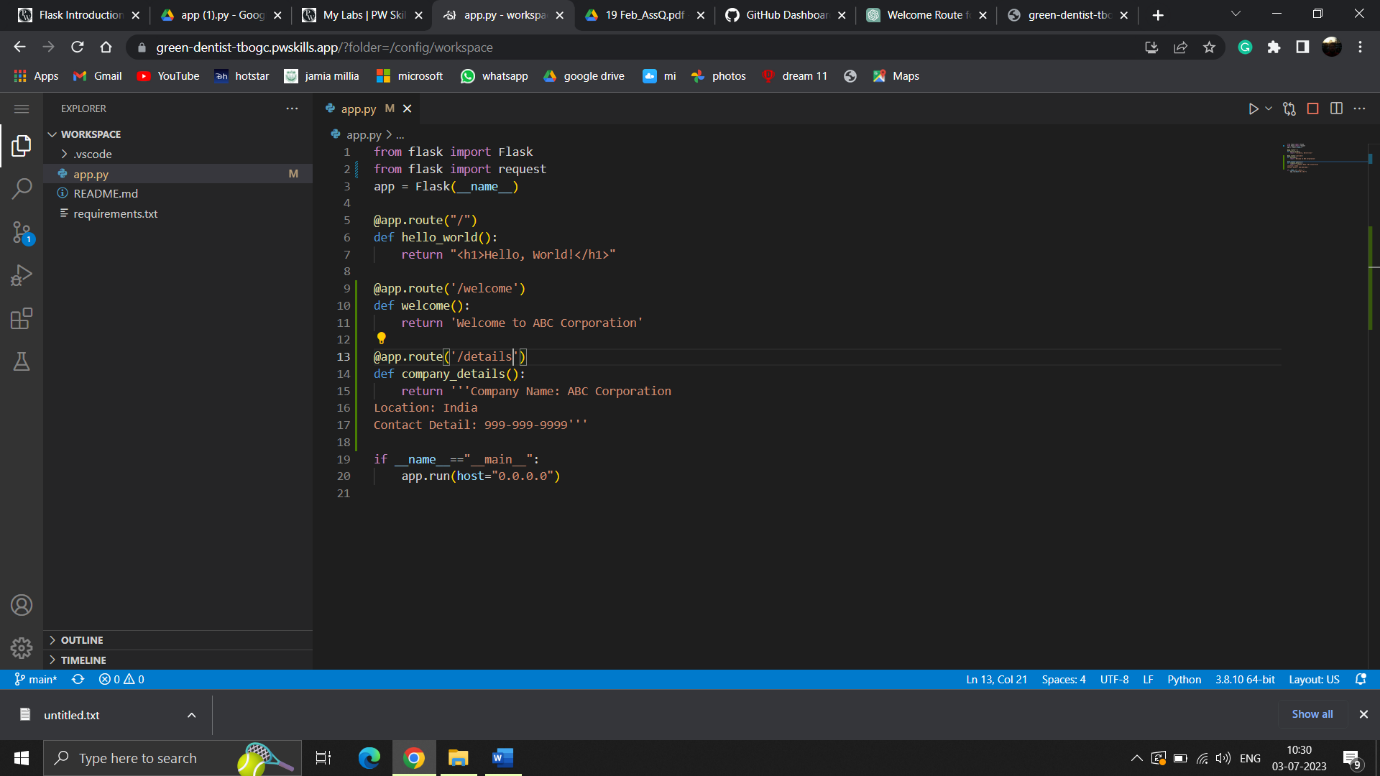
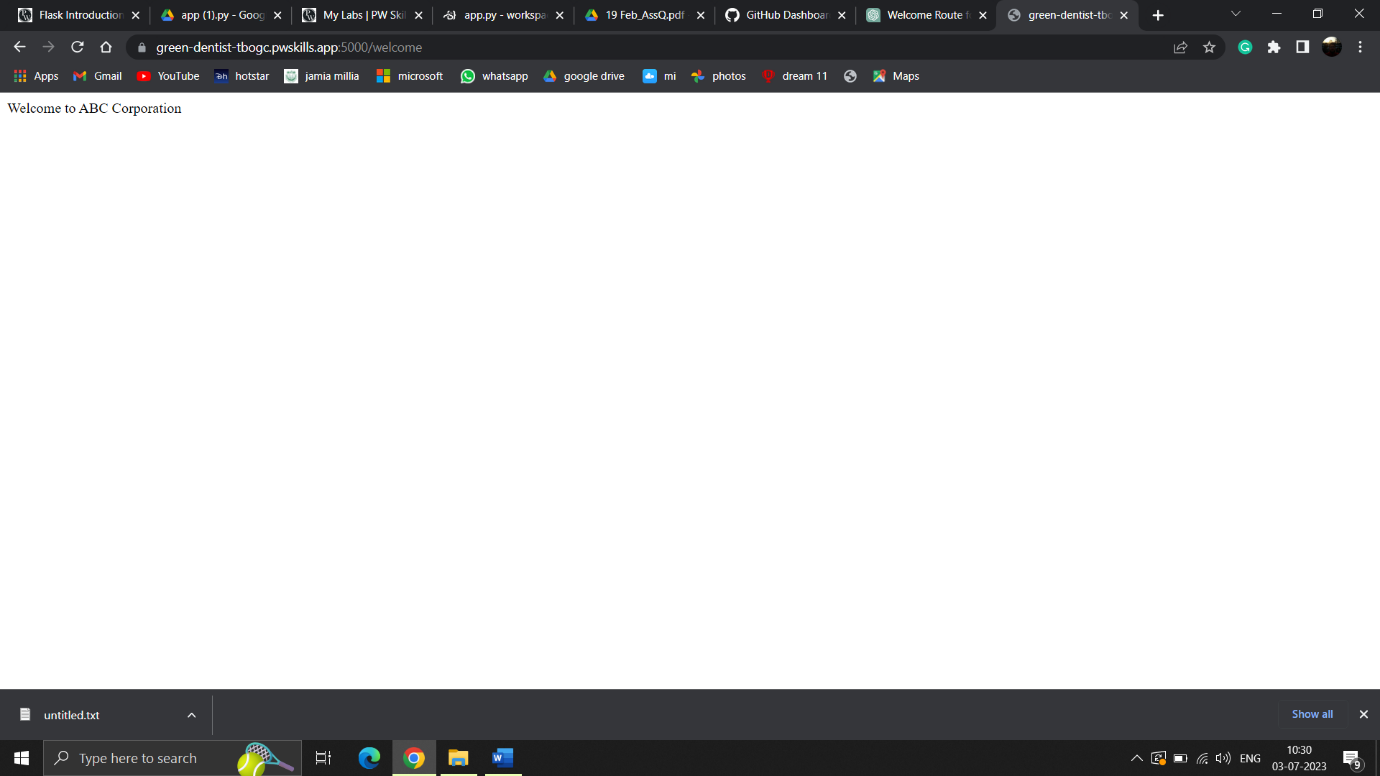
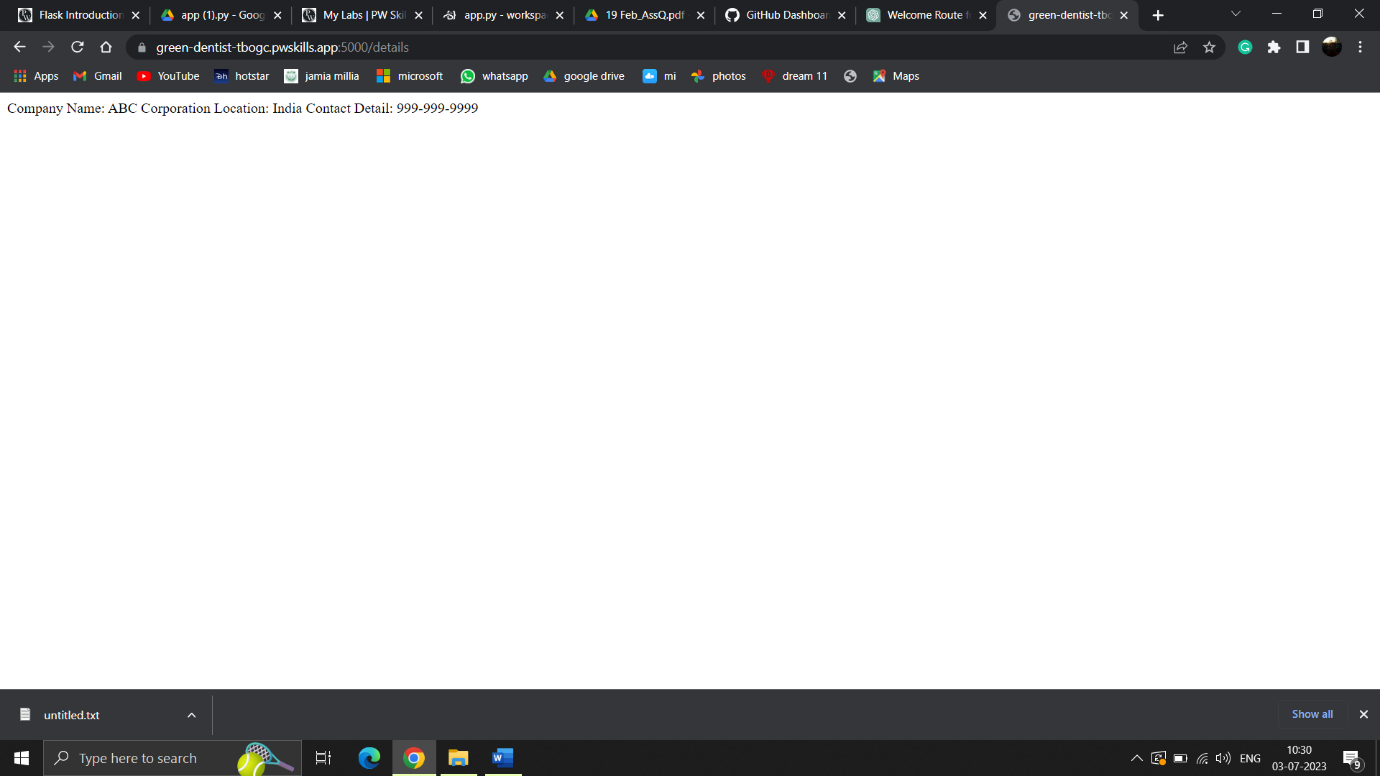
#By mapping routes to functions, we can specify what should happen when a user visits a particular URL.

#For example, we can define a route to display a specific page, process form data, or return JSON responses for an API.

#App routes provide a clean and organized way to structure the application's functionality.

#They allow developers to handle different requests separately and make the code more maintainable and modular.

#Additionally, app routes help in building RESTful APIs by defining routes for various resources and specifying the appropriate HTTP methods to handle them.

Q4

Q5

#In Flask, the url\_for() function is used for URL building.

#It helps generate URLs for a given route function by taking into account the route's name and any parameters it requires.

