****

**Name of the Student: Choudhari Arti Rajendra no: 08**

**CLASS:- B.E.[I.T] Division: ACourse:- 2015**

**Assignment No.7**

**COMPUTER LABORATORY-IX**

**Marks: /10**

**Date of Performance: Sign with Date:**

## ASSIGNMENT NO.7

**ProblemStatement:**

To develop micro services framework based distributed application.

**Objective:**

1) The course aims to provide an understanding of the principles on which the distributed systems are based; their architecture, algorithms and how they meet the demands of Distributed applications.

2) The course covers the building blocks for a study related to the design and the implementation of distributed systems and applications.

**Outcomes:**

1) Demonstrate knowledge of the core concepts and techniques in distributed systems.

2) Learn how to apply principles of state-of-the-Art Distributed systems in practical application.

**PEOs:2; POs: a,b,c,d,e,f,f,g,i, l, m ; PSOs: 1,2,3 and COs satisfied: 1, 2.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assignment No.** | **Assignment Title** | **Assignment Statement** | **Scenarios** | **Software Required** |
| 7 | Microservices | To develop Microservices framework based distributed application. | - | Study Assignment |

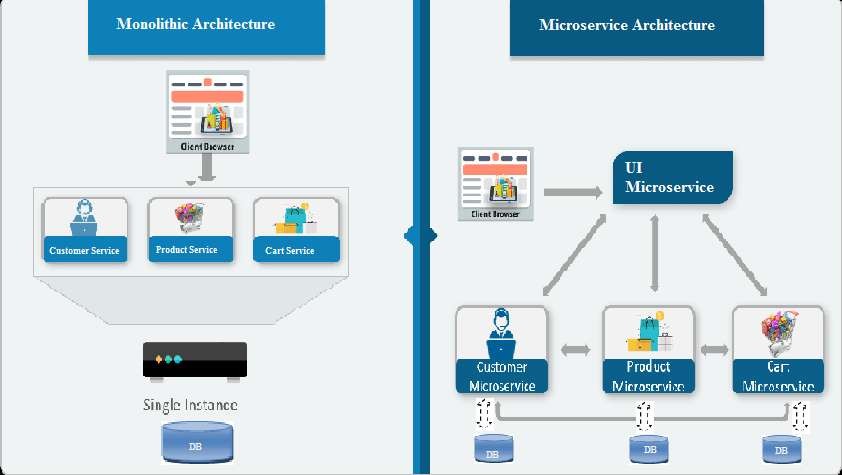
## Tools /Environment:

Python 3.6.0 using Flask framework.

## RelatedTheory:

1. **Microservices:**

Traditional application design is often called “monolithic” because the whole thing is developed in one piece. Even if the logic of the application is modular it’s deployed as one group, like a Java application as a JAR file for example. This monolith eventually becomes so difficult to manage as the larger applications require longer and longer deployment timeframes. In contrast with the monolith type application, here’s what an app developed with a microservices focus might looklike:



A team designing micro services architecture for their application will split all of the major functions of an application into independent services. Each independent service is usually packaged as an API so it can interact with the rest of the application elements.

Microservices - also known as the microservice architecture - is an architectural style that structures an application as a collection of services that are:

* + Highly maintainable andtestable
  + Looselycoupled
  + Independentlydeployable
  + Organized around businesscapabilities.

The microservice architecture enables the continuous delivery/deployment of large, complex applications. It also enables an organization to evolve its technology stack.

1. **Web frameworks** encapsulate what developers have learned over the past twenty years while programming sites and applications for the web. Frameworks make it easier to reuse code for common HTTP operations and to structure projects so other developers with knowledge of the framework can quickly build and maintain theapplication.

**Common web framework functionality:** Frameworks provide functionality in their code or through extensions to perform common operations required to run web applications. These common operations include:

1. URLrouting
2. Input form handling and validation
3. [HTML,](https://www.fullstackpython.com/hypertext-markup-language-html.html) XML, JSON, and other output formats with a [templatingengine](https://www.fullstackpython.com/template-engines.html)
4. Database connection configuration and persistent data manipulation through an [object- relational mapper (ORM)](https://www.fullstackpython.com/object-relational-mappers-orms.html)
5. [Web security](https://www.fullstackpython.com/web-application-security.html) against Cross-site request forgery (CSRF), SQL Injection, Cross-site Scripting (XSS) and other common maliciousattacks
6. Session storage and retrieval.
7. **Flask** (source code) is a Python web framework built with a small core and easy-to-extend philosophy. Flask is based on the Werkzeug WSGI toolkit and Jinja2 templateengine.
8. **WSGI:** Web Server Gateway Interface (WSGI) has been adopted as a standard for Python web application development. WSGI is a specification for a universal interface between the web server and the web applications.
9. **Werkzeug :**It is a WSGI toolkit, which implements requests, response objects, and other utility functions. This enables building a web framework on top of it. The Flask framework uses Werkzeug as one of itsbases.
10. **VirtualEnvironment:**

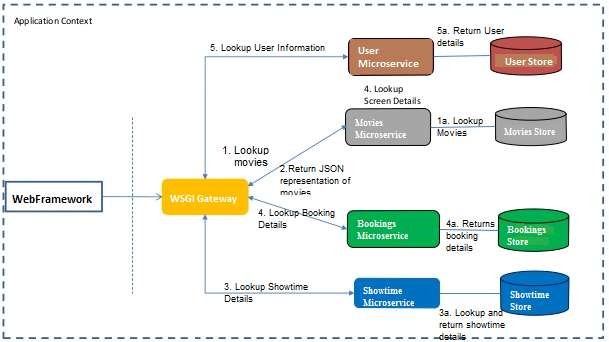
In Python, by default, every project on the system will use the same directories to store and

retrieve**site packages** (third party libraries). and**system packages** (packages that are part of the standard Python library). Consider the a scenario where there are two projects: *ProjectA*and*ProjectB*, both of which have a dependency on the same library, *ProjectC*. The problem becomes apparent when we start requiring different versions of *ProjectC*. Maybe *ProjectA*needs v1.0.0, while *ProjectB*requires the newer v2.0.0, forexample.

Since projects are stored in site-packages directory according to just their name and can't differentiate between versions, both projects, *ProjectA*and*ProjectB*, would be required to use thesameversionwhichisunacceptableinmanycasesandhencethevirtualenvironment.The

main purpose of Python virtual environments is to create an isolated environment for Python projects. This means that each project can have its own dependencies, regardless of what dependencies every other project has. There are no limits to the number of environments you can have since they’re just directories containing a few scripts. Plus, they’re easily created using the virtualenv or pyenv command linetools.

## Designing thesolution:



Here, we are attempting to develop anmicroservice based architecture for Movie ticket Booking web application. The services are being implemented using python and JSON is used as for Data Store.

## Implementing thesolution:

1. **Using Virtual Environments:** Install virtualenv for development environment. virtualenvis a virtual Python environment builder. It helps a user to create multiple Python environments side-by-side. Thereby, it can avoid compatibility issues between the different versions of thelibraries.

The following command installs virtualenv: Sudo apt-get install virtualenv

## Flask Module:

Importing flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of current module (name) as argument. The route() function of the Flask class is a decorator, which tells the application which URL should call the associated function.

## Route decorator:

The route() decorator in Flask is used to bind URL to a function. For example −

@app.route(‘/hello’) defhello\_world():

return ‘hello world’

Here, URL ‘/hello’ rule is bound to the hello\_world() function. As a result, if a user visits http://localhost:5000/hello URL, the output of the hello\_world() function will be rendered in thebrowser.

1. **Writing the subroutine for the four microservices:** There are four microservices viz.,user, Showtimes, Bookings and Movies for which microservices are to beimplemented.

**Conclusion:** With microservices, modules within software can be independently deployable. In a microservices architecture, each service runs a unique process and usually manages its own database. This not only provides development teams with a more decentralized approachto

building software, it also allows each service to be deployed, rebuilt, redeployed and managed independently. Netflix, eBay, Amazon, the UK Government Digital Service, Twitter, PayPal, The Guardian, and many other large-scale websites and applications have all evolved from

monolithic to microservices architecture.