

Indian Institute of Technology, Delhi

ELL887 : Cloud Computing

Assignment 2

**Name:** Himanshu Singh Purte

**Entry No.:** 2023EET2179

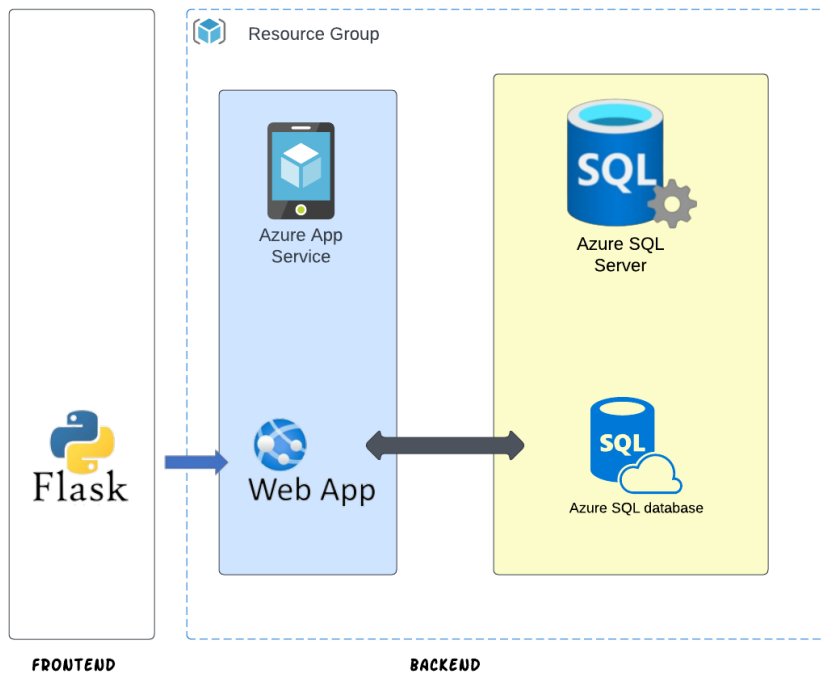
**Course Instructor:** Prof. Sougata Mukherjea

# Contents

<b>1</b>	<b>Architecture Diagram</b>	<b>1</b>
<b>2</b>	<b>The Architecture Decisions</b>	<b>1</b>
2.1	Flask . . . . .	1
2.2	Pyodbc . . . . .	1
2.3	Azure Resource Group . . . . .	1
<b>3</b>	<b>WebApp URL</b>	<b>2</b>



## 1 | Architecture Diagram



## 2 | The Architecture Decisions

### 2.1 | Flask

Python with Flask is a potent combination for web development, offering a lightweight and easy-to-use framework for building applications and APIs. Flask simplifies routing with decorators, supports dynamic HTML rendering, and enables the handling of static files effortlessly. It excels in RESTful API development, provides request and response objects for efficient data management, and boasts a rich ecosystem of extensions for added functionality. With Flask, developers can rapidly prototype and deploy applications, leveraging Python's readability and simplicity to create scalable and versatile web solutions.

### 2.2 | Pyodbc

PyODBC is a Python library that provides an interface to ODBC (Open Database Connectivity). We are using ODBC 18 for this assignment. Allowing Python programs to connect to and interact with databases using standard SQL queries. With PyODBC, developers can connect to a wide range of database management systems (DBMS) including Microsoft SQL Server, PostgreSQL, MySQL, Oracle, and others, making it a versatile tool for database interaction in Python applications. PyODBC simplifies database access by providing a consistent API across different database platforms, enabling tasks such as executing SQL queries, fetching query results, and managing database connections. It's widely used in various industries for data-driven applications, data analysis, and integration with existing database systems, offering flexibility, performance, and ease of use for Python developers.

### 2.3 | Azure Resource Group

A Python web application deployed in Azure that utilizes Azure Web Services and SQL Database represents a powerful combination for building scalable, reliable, and cloud-native web solutions. Leveraging Azure's robust infrastructure and services, developers can create dynamic web applications with seamless integration between the application logic, data storage, and cloud-based hosting environment.



The web application, built using a Python web framework Flask , interacts with **Azure Web Services**, which provides a platform for hosting web applications with features such as auto-scaling, load balancing, and continuous deployment. This allows developers to focus on building and deploying their application code without worrying about managing the underlying infrastructure.

For data storage, the application utilizes **Azure SQL Database**, a fully managed relational database service offered by Microsoft Azure. Azure SQL Database provides high availability, security, and scalability, allowing developers to store and manage their application data in the cloud with ease. The Python application communicates with Azure SQL Database using libraries such as pyodbc, enabling seamless integration with the database backend.

By deploying a Python web application in Azure that utilizes Azure Web Services and SQL Database, developers can take advantage of the scalability, reliability, and performance of the Azure cloud platform, enabling them to build and deploy modern web applications with confidence. This architecture offers flexibility, cost-effectiveness, and streamlined management, making it an ideal choice for a wide range of web development projects.

### 3 | WebApp URL

You can visit the website at [nick232179.azurewebsites.net](http://nick232179.azurewebsites.net).