

# Using Python Flask Created “Disease Detector WebApp”

[Azure WVD, Virtual Machine, Azure Web App Service]

**Problem statement –**

**Solution -**

**Let’s look into technical details and the Implementation of the solution:**

**You need the following software/ Azure Account, please find details below**

**Azure Account:- get an azure account by clicking on the following link**

<https://azure.microsoft.com/en-us/free/>

**There is free credit for students and 200 USD credit if you want to get started with Azure**

<https://azure.microsoft.com/en-us/free/students/>

**Visual Studio 2019 Community**

<https://visualstudio.microsoft.com/downloads/>

**Community edition is free for students and open-source contributors for non-commercial use.**

**Visual Studio Code (Optional if you have Visual Studio 2019 for Azure Function Development)**

<https://code.visualstudio.com/download>

## **Introduction to Azure VM and MongoDB Atlas on Azure**

Microsoft Azure is a cloud computing platform where we can host our applications, websites, and other services. Azure has a host of services, including hosting and maintenance, database management, and app development. MongoDB Atlas on Azure is a fully managed database service that saves you the hassle of managing your own database infrastructure. It delivers high performance and scales as your application grows. It also provides the full functionality of the open-source version of MongoDB with a click of a button.

## **Basics of MongoDB/Atlas: Why it is Important for Modern Crop Marketing System?**

Modern Crop Marketing System is a complex system and to build such a complex system we need to use various tools such as a database, a programming language, and an operating system. Here in our case, we are going to use a database and a programming language, but what about an operating system? In this case, we will use Azure VM to host our application because the VM has an operating system that will host our application. We can use any programming language and database if we host our application in an Azure VM. It means that we can use any database and any programming language in our application because we have full-fledged operating systems. In our application, we will use MongoDB as a database because it is easy to use and scalable. This means that the data in MongoDB is easy to update and it can scale. That is why we use MongoDB as a database.

## **Pre-requisite for using JavaScript Deployed Modern Crop Marketing System with Azure VM, MongoDB Atlas on Azure, and Azure Web App.**

In order to use our JavaScript-deployed Modern Crop Marketing System with Azure VM, MongoDB Atlas on Azure, and Azure Web App, first of all, you need to have an Azure VM and MongoDB Atlas account.

## **Steps to Create and Connect Azure VM and MongoDB Atlas:**

**Step 1: Create an Azure VM and deploy the app**

Before deploying the app and connecting it to the database, let's first create an Azure VM for hosting the app. Launch the Azure Portal, select New, and then select App Services. Select Hosted service, select Web App, provide a Name, select the Subscription and Location, and then click Create.

## **Step 2: Install MongoDB on Azure VM and update the connection string**

The Azure VM has a virtual machine that can be accessed from any computer with the help of a secure connection. There are two ways to install MongoDB Enterprise on Azure VM: We will use the second method because the former method does not allow you to update the database version for any reason. Now, let's connect to Azure VM. Go to Azure VM and secure the connection by generating an SSH key and adding it to the VM. After that, it will prompt a command to run on the VM. Now, let's install MongoDB on the VM. `MongoDB-org-install-Linux-arm.sh`, After installation, the database can be accessed with the help of the mongo shell. `mongo` After the database is accessed, we need to update the connection string with the database IP and port.

## **Step 3: Create a web app and import data from MongoDB Atlas**

Let's create a web app in Azure to connect to the database and import data. First, go to the Azure Portal and select New. Select App Services and then select Web App. Provide a Name, select the Subscription and Location, and then click Create. After creating the web app, go to the Dashboard, select Settings, and then select Service Endpoints. Now, click Add New and select the MongoDB connection. After that, provide the MongoDB IP and the port used to access the database. Click Save.

## **Step 4: Set up Azure Functions for Marketing Actions (Email, SMS, etc.)**

Now, let's create Azure Functions for email, SMS, marketing automation, etc. For that, go to the Azure Portal and select New. Select App Services, select Function App, select the Subscription and Location and then click Create. Click on Functions, select Create New, provide a Name, select Function App and then select the Location, select the Language, provide a Function Name, and then click Create.

## **Summary**

This JavaScript-deployed Modern Crop Marketing System demonstrates how to build a scalable and secure serverless architecture in Azure with MongoDB

**Atlas, Azure Functions, and Azure VM. To create and connect Azure VM and MongoDB Atlas, follow these steps - create an Azure VM, install MongoDB on Azure VM, create a web app, and then import data from MongoDB Atlas. These steps will guide you through the process of creating a scalable and secure serverless architecture in Azure.**