

1. PROJECT TITLE:

Student Portal for College/Institute

2. PROBLEM STATEMENT/OPPORTUNITY:

XYZ College provide various vocational and industry specific skill courses at diploma and undergraduate level. As of now all their student services, starting from course enquiry, admission process, class schedules, Fee status etc. are handled manually over the enquiry counter. During the normal course Only one enquiry counter suffices the purpose however during the peak times (session start month) college administration has to arrange for 2-3 such counters to handle all such services. College management is also planning to expand its operations/services and open two more campuses in the neighboring states. Objective of this project is to provide a self-service portal automating some of these processes and lay a foundation to cover the rest gradually.

3. PROJECT DESCRIPTION:

The solution will involve building a Web portal which will provide information to two sets of users.

1. Students who are already registered and pursuing their courses at the institute.
2. The visitors like prospective students/media/education counsellors etc.

The registered users(students) will have access to services such as class schedules, exams schedule, fees dues, attendance records, transcripts etc.

The unregistered users will only have access to the general website, which has the frequently asked questions, course details, admission process, news, general publications, webinars and other information about the college.

The users of the web app will have the option to subscribe to the college's newsletter.

For a social cause, portal will provide features through which students can publish their volunteer services as a tutor and the other external/internal students can put a request to avail such services without any charge.

To build a scalable solution and in future take advantage of new digital capabilities like AI etc., we will design and implement this solution on Cloud making use of its cost effective and robust services.

4. Features

The Web App has the following features:

- BOT: Web app has a bot which is designed to help visitors on the website by answering FAQs.
- The Web App has a student portal on which the registered students will be able to query information like results, fee dues, class schedules, attendance records etc. from college information systems.

5. How to test (see demo)

- 1) Click on the below URL to access this web application.

<http://xyzapp.centralindia.cloudapp.azure.com:8080/home>

This will show XYZ college home page.

NOTE: TO SAVE COST THE PROJECT DEMO MACHINE WILL BE RUNNING ONLY BETWEEN 10AM-6PM IST MON-FRI.

- 2) TO access Student portal

In order to access the Student Portal you may use following Test user credentials :

USER NAME	PASSWORD
Raj	1_r
Betty	2_b

3) After successful authentication you will be directed to the Student Portal.

In case you face an issue in accessing the App please write to: tanejanandika@gmail.com for support.

6. Dependencies

The following infrastructure and software components are used for this project:

- 1) Azure
 - I. VM (to host Web App workload) with following components: -
 - a. Linux (ubuntu 20.04)
 - b. mysql-client-core-8.0
 - c. Python version:3.8
 - i. Virtual environment
 - ii. Django
 - iii. mysql-client
 - iv. mysql-connector-python
 - d. git
 - II. Azure Database for MySQL flexible server
 - a. MySQL version 5.7
 - III. Web app BOT
 - a. QnA maker
 - i. Knowledge base
 - Create FAQs
 - Import FAQs
 - b. Knowledge base
 - IV. Function App (to automate start/stop of VM and Database at pre-defined time slots)
 - a. Operating system-Windows
 - b. PowerShell
 - i. Az module version 7.*
 - V. Persistent Storage for VM
 - VI. Microsoft.Compute/images
- 2) Repository: GitHub

7. Setup / Installation Steps

- 1) Login to Azure portal with developer access
- 2) Set up Database
 - I. Azure Database for MySQL flexible server with following parameters: -
 - i. Name: ntddb (this name is used in making connection from app to database)
 - ii. private IP address
 - iii. MySQL version: 5.7
 - iv. Compute size: Standard_B1s
 - v. Storage:20 GiB
 - vi. Virtual network: prj1-vnet/ntsubnet
 - II. Set up db model and sample data
 - i. Connect to database using MySQL client
 - ii. Create database xyz_college.
 - iii. Load database and sample data using following command:
 1. `mysql -u username -p new_database < dbdump.txt`

3) Set up VM and Project files:

I. Create Azure VM

- i. From Azure portal create a VM based on following parameters: -
 1. Image Ubuntu Server 20.04
 2. Set Username and password to access the VM
 3. Enable inbound ports(HTTP, SSH,HTTPS)
 4. Size: Standard B1s
 5. vCPUs:1
 6. RAM:1 GiB
 7. Virtual network: prj1-vnet/ntsubnet
- ii. In properties set DNS name (xyzapp)
- iii. Note the VM's public IP address to be used in Putty

II. Install app on VM

- i. Open SSH using putty to VM(use the IP address captured above)
- ii. In the VM CLI enter the VM username and password
- iii. `sudo apt-get install python3`
- iv. `sudo apt-get install pip`
- v. `sudo pip install virtualenv`
- vi. `virtualenv venv`
- vii. `source ./venv/bin/activate`
- viii. `cd venv`
- ix. `pip install django`
- x. `pip install mysql-client`
- xi. `sudo pip install mysql-connector-python`
- xii. `sudo apt-get install git`
- xiii. Download project from git hub- git clone command
- xiv. Modify settings.py to update database connection settings as per the DB setup above viz. Username, Password, DB HOST.
- xv. Connect to MySQL database using MySQL client

NOTE: AFTER DOING STEPS 1-2 create a VM image which can be reused to create more VM's if required.

- xvi. Create a script to automate application startup when VM starts:
 1. Copy xyz.service to this location `/etc/systemd/system/`
 2. Activate the service using the following command:
 - a. `systemctl enable xyz.service`

4) Create Function App (sample files included in "AzureAppFiles/xyzc.zip")

I. Operating System: Windows

II. Create the following functions with time trigger template: -

- i. TimeTrigger1 (event used to stop VM and Database)
 1. Modify PowerShell script to include commands to STOP VMs (see example: TimerTrigger1\run.ps1)
- ii. Time Trigger2 (event used to start VM and Database)
 1. Modify PowerShell script to include commands to STOP VMs (see example: TimerTrigger2\run.ps1)