

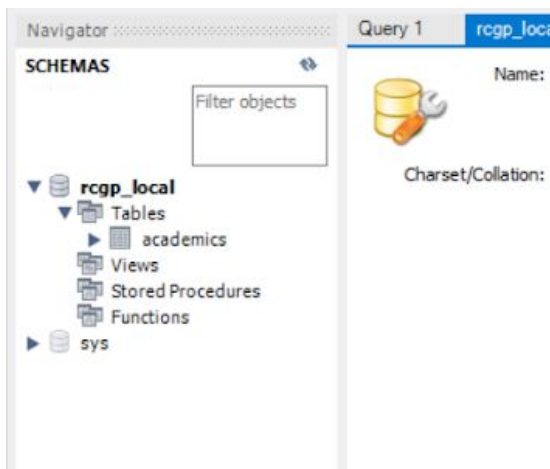
Offline Deployment Guide

Note: This is for offline deployment only. The chatbot can also be found hosted online at azure here: <https://chatbotdeployment.z22.web.core.windows.net/>

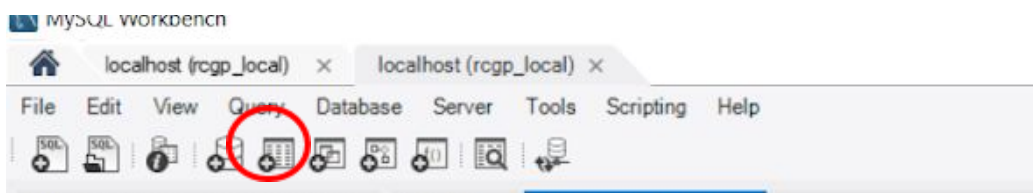
Pre-requisites:

- **Python 3.7 with the following libraries**
 - mysql-connector-python
 - fuzzywuzzy
 - tkinter
- **Microsoft Bot Framework Emulator**
- **.NET Core**
- **MySQL Workbench**
- **MySQL Server**

1. First of all, a local MySQL server needs setting up. To do so, follow this link for step by step instructions: <https://ladvien.com/data-analytics-mysql-localhost-setup/>
2. Once you have setup the server, run it and note down the hostname, database name, username and password.

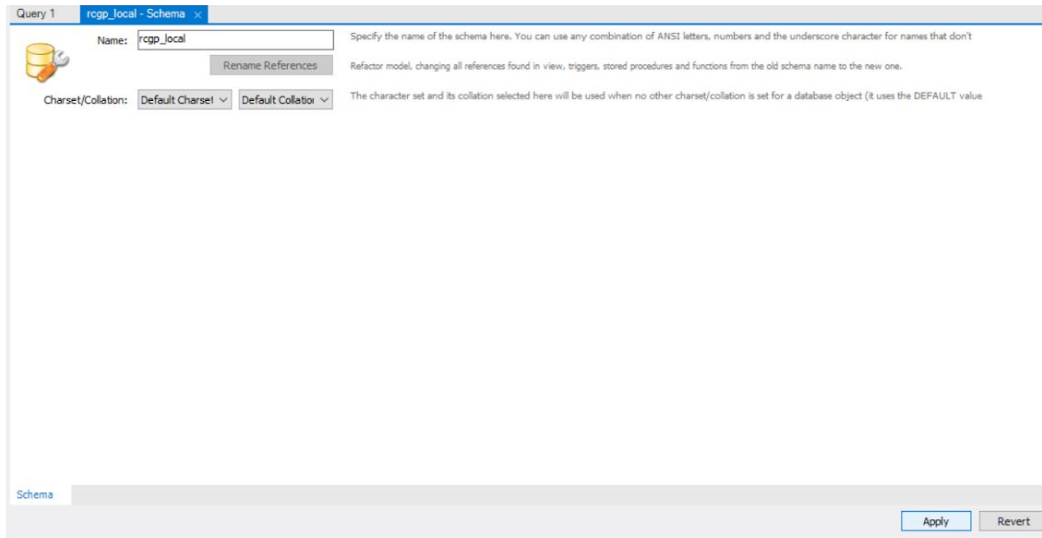


3. Open MySQL workbench and connect to the database you just created. If you already have a database schema (should be listed on the left-hand side of the workbench as shown above), skip to step 5. If not, go to the taskbar and click on the icon that says 'Create a new schema in the connected server' (shown below)

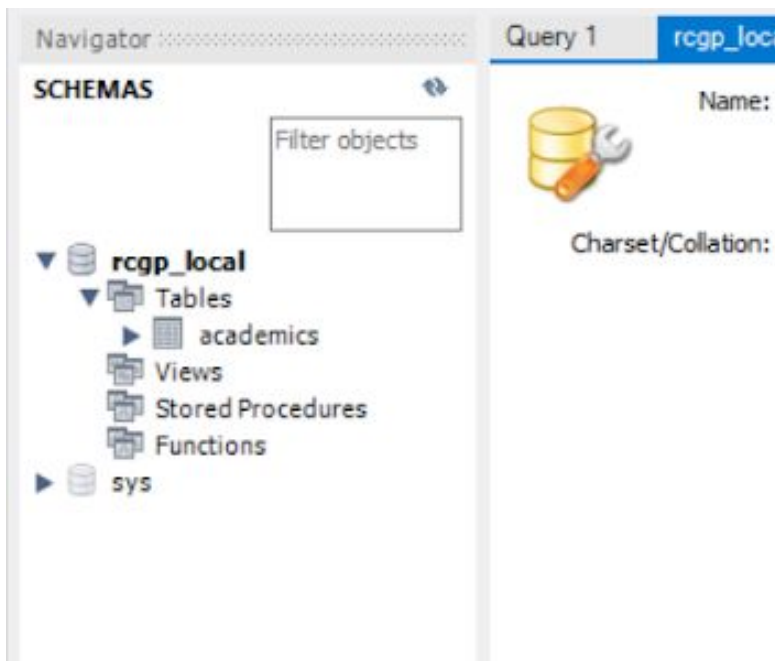


Offline Deployment Guide

- Now give a name to the schema (we suggest 'rcgp_local' as that is what we used) and click apply in the bottom right hand corner.



- Your schema should now appear on the left-hand side. Double click it so that it is highlighted in bold.



- Now navigate to the 'deployment' folder and open up the 4 scripts needed to repopulate the database.
- Execute the scripts in the order 'rcgp_projects_academics', 'rcgp_projects_contracts', 'rcgp_projects_projects' and finally 'rcgp_projects_projectentries'
- To check whether step 4 was successful, you should be able to see 4 new tables in your database and perform selection queries on each to see data populated in them.

Offline Deployment Guide

9. Once this has been setup, navigate to the directory 'Projects Chatbot/ixnChatbot/Database/' and open the JSON file dbconfig.json'. Enter the database settings required for your MySQL server. The default settings provided should map to a localhost server already. Database and password should be the only thing that requires changing. Note: Server refers to the hostname.

(see below for what the file looks like)

```
{
  "server": "127.0.0.1",
  "database": "rcgp_local",
  "username": "root",
  "password": ""
}
```

10. Now, navigate to the directory 'ixnMiner/config.json' and once again, enter the database information. The only information that should be required is the server (hostname), database, username and password. The server and username should already be filled out for a local server.

DO NOT TOUCH entrytable, projectstable, contractstable or academicstable. These are fixed table names that cannot change.

```
11. {
  "server": "127.0.0.1",
  "database": "rcgp_local",
  "entrytable": "projectentries",
  "projectstable": "projects",
  "contractstable": "contracts",
  "academicstable": "academics",
  "username": "root",
  "password": ""
}
```

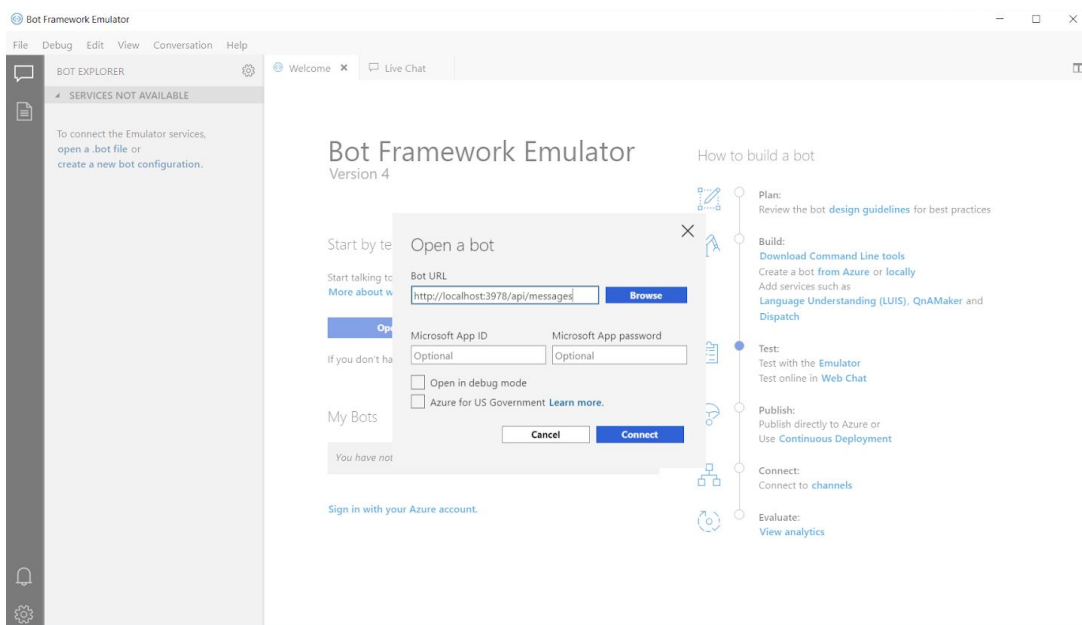
12. Navigate to 'Projects Chatbot/ixnchatbot' and open up a terminal. Type in 'dotnet run' and press enter to run the chatbot. You should see a URL displayed showing where the web app is being hosted.

Offline Deployment Guide

```
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.

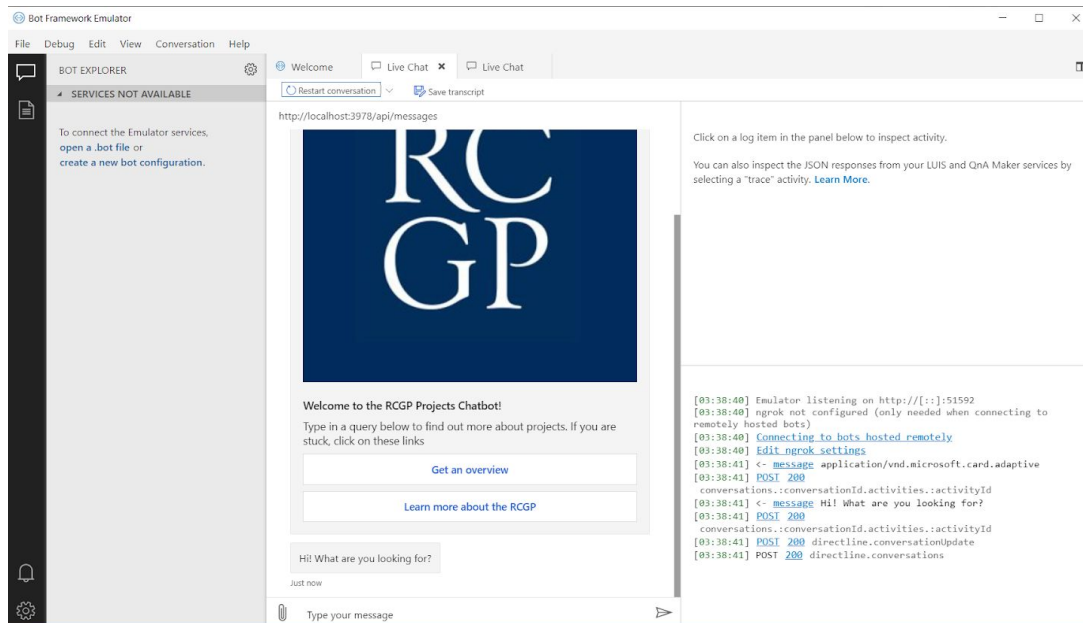
C:\Users\adnan\OneDrive\Documents\Uni\Second Year\System Engineering\42_code\Projects Chatbot\ixnChatbot>dotnet run
info: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[0]
      User profile is available. Using 'C:\Users\adnan\AppData\Local\ASP.NET\DataProtection-Keys' as key repository and
Windows DPAPI to encrypt keys at rest.
Hosting environment: Development
Content root path: C:\Users\adnan\OneDrive\Documents\Uni\Second Year\System Engineering\42_code\Projects Chatbot\ixnCha
tbot
Now listening on: https://localhost:3979
Now listening on: http://localhost:3978
Application started. Press Ctrl+C to shut down.
```

13. Now open up the Microsoft Bot Emulator and click the 'open bot' button. Copy and paste the URL listen in your console with '/api/messages' appended to it. (see below)



14. Click connect and you should see the chatbot open up and display a welcome message.

Offline Deployment Guide

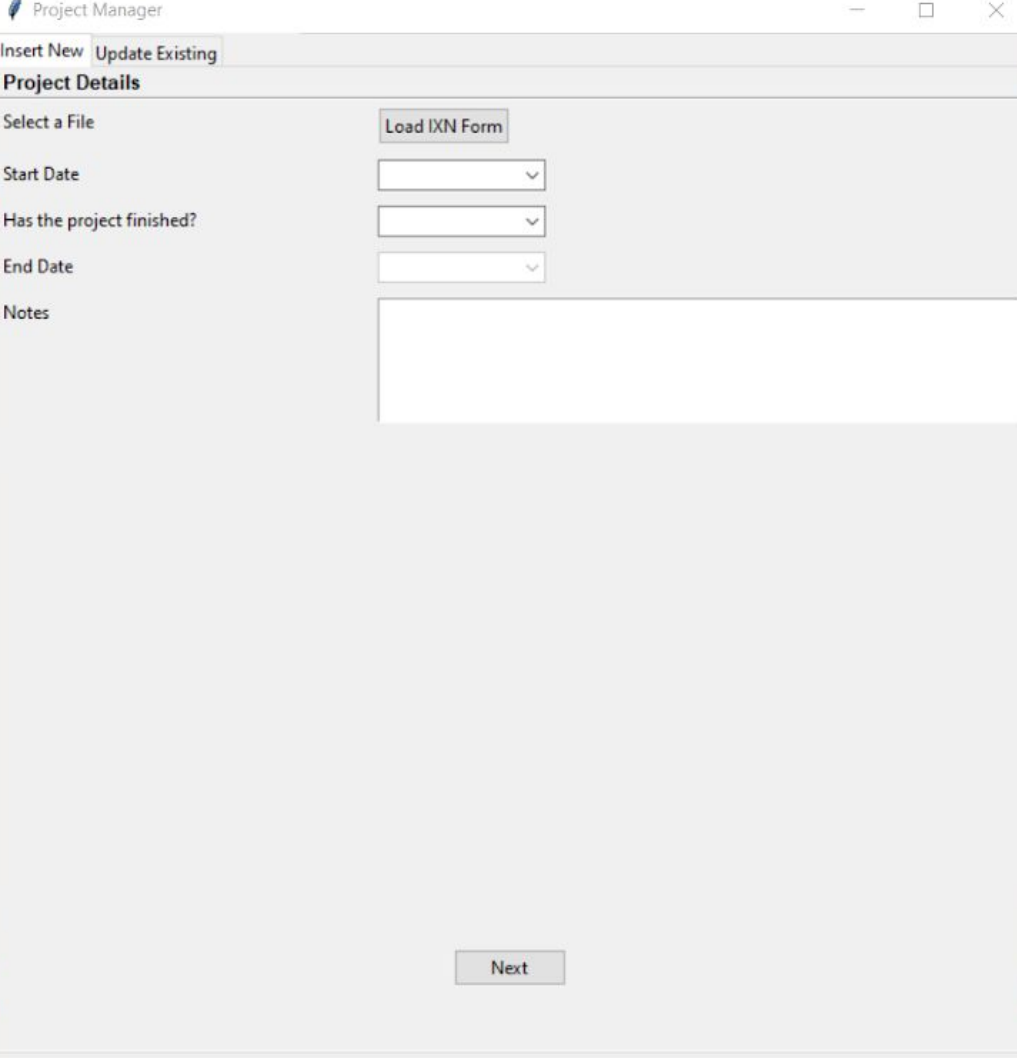


15. The chatbot can now be queried with the existing projects in the database. For help with what to do, type in 'help' and view instructions.

Offline Deployment Guide

Administration Panel

1. The administration panel should also be setup to work now. Run the miner.py file and the GUI interface should open. Any IXN forms can then be pushed to the project database. A sample projects folder named 'projects' has been included in this submission to test this with.



The screenshot shows a window titled "Project Manager" with standard Windows window controls (minimize, maximize, close). Inside the window, there are two tabs: "Insert New" and "Update Existing", with "Update Existing" currently selected. Below the tabs is a section titled "Project Details". This section contains several input fields and buttons:

- A "Select a File" label followed by a "Load IXN Form" button.
- A "Start Date" label followed by a date selection dropdown menu.
- A "Has the project finished?" label followed by a dropdown menu.
- An "End Date" label followed by a date selection dropdown menu.
- A "Notes" label followed by a large, empty text area for input.
- A "Next" button located at the bottom center of the form area.