**US\_VISA\_APPROVAL\_PREDICTION**

**Project Setup**

* Anaconda Installation
* VS code
* Git hub project repository creation

1. **Project template creation**
   * template.py
     + Creating the required folders and files
   * Git commands
     + git add .
     + git commit -m “template created”
     + git push origin main
   * **\_\_init\_\_** wherever it is can be considered as a local package
2. **Create a virtual environment**
   * Install environment: conda create -n visa python=3.8 -y
   * activate environment: conda activate visa (environment name: visa)

==> WARNING: A newer version of conda exists. <==

current version: 23.7.4

latest version: 24.5.0

Please update the conda by running.

$ conda update -n base -c defaults conda

Or, to minimize the number of packages updated during the conda update, use

conda install conda=24.5.0

1. **Install Requirements** 
   * pip install -r requirements.txt
   * **setup.py (**-e .**)**
2. **MongoDB\_connection**
   * MongoDB login
   * Project create
   * Connect to the database server
   * Create login credentials
   * Using Python client to add data to the database
   * Get the connection string

mongodb+srv://naveen0383:naveen0383@cluster0.lhrniw3.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0

* + Insert data into the database

1. **Logging**
   * Trace the file that you are running and get the time stamp at different logger levels
2. **Exception Handling**
   * Check whether there are any errors when executing the code.
3. **Utils**
   * For reading config. yaml file
   * Store functions and utilize them whenever it is required.

**Creating End-to-End pipeline**

1. **Notebook experiments**
   * Extrapolatory Data Analysis (EDA)
   * Model Training
   * Model Performance Evaluation
2. **Constants**
   * Can be used everywhere whenever required. Common for the project.
3. **Data Ingestion component**
   * Data Ingestion from MongoDB database
   * MongoDB connection with the connection string
   * Read the data from the Collection and convert it into a data frame
4. **Data Ingestion**
   * Config entity
   * Artifact entity
   * Data ingestion
5. **F**
6. **D**