**Do LLM’s (Large Language Model) understand human logic? If so, at what level? Can LLM’s understanding of human logic be improved?**

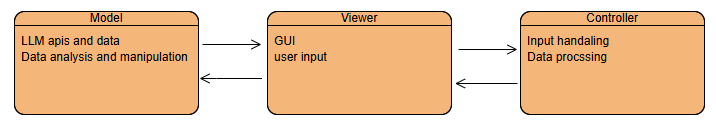
**Architecture**- The architecture best suits our project is MVC (Model , View , Controller).

* **Model:**
  + Hold api keys to the LLM, questions ,answers prompts and scores
  + Handle data retrieval,storage and processing
* **view:**
  + GUI created using Tkinter
  + Accepts user input.
* **controller:**
  + Acts as a bridge between the Model and the View, mostly automated by Tkinter.
  + Handles user input, processes it, and updates the questions and view accordingly.

**Data Storage**

We Will store the prompts in csv file, scores, question and answers

**Graphic description**

****

|  |
| --- |
| **sae all the api**   * **manage al the** |

**Data Description**

* **Prompts**
  + what type of logical question the prompt should help in(Propositional Logic,Non-monotonic Reasoning,First order logic)
  + Instruction to help the model with the type of logical questions
* **Question**
  + Type of question (Propositional Logic,Non-monotonic Reasoning,First order logic)
  + Type of answer (binary/multi choice/open ended question
  + The question
  + The answer to the question
* **Success Metrics** 
  + to compare the user experience with the system to our study we will save our scores and present them against his
* **LLM’s**
  + Name of the model and it version
  + API key (we will use it to connect our model to the LLM)

**Programming Languages and Tools**

For the app we will use Python, to show graphs we will use Matplotlib and Seaborn, For numerical calculation we will use Numpy and for the GUI we will use Tkinter(probably)

**User interface:**

* Choose LLM
* Choose type of logical question
* Choose Prompt
* Show our scores
* Compere results
* Upload a question
  + Upload answer for that question

