



# **Summer Research Internship**

## **Visualizing Biomedical Database using Underlying Knowledge Graph**

**Supervisor: Dr. Jayanta Mukhopadhyay**

**Mentor: Ankita Saha**

**Contributor: Yashi Kesarwani (IIIT Kalyani)**



# Pipeline

- **Knowledge Graph Handling Creation**
- **Data Visualization**
- **Flask API**
- **User Interface for input and required visualization**



# Technologies Stack and Dependencies

## For Knowledge Graph Handling Creation

- Python
- Py2neo and Neo4jRestClient Framework
- Neo4j Localhost Server

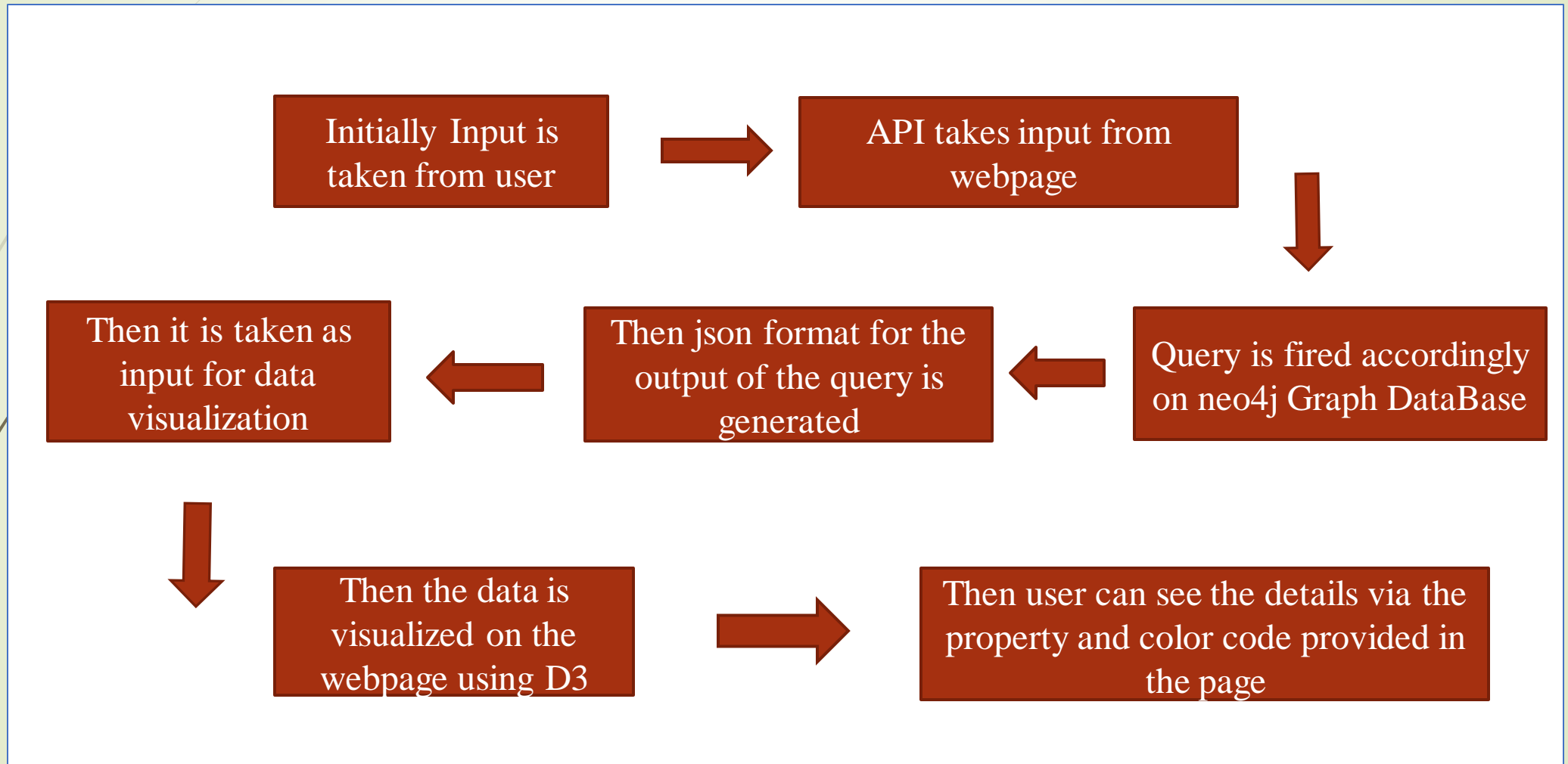
## For Data Visualization

- Data Driven Documents (D3)
- Scalar Vector Graphics

## For API and User Interface

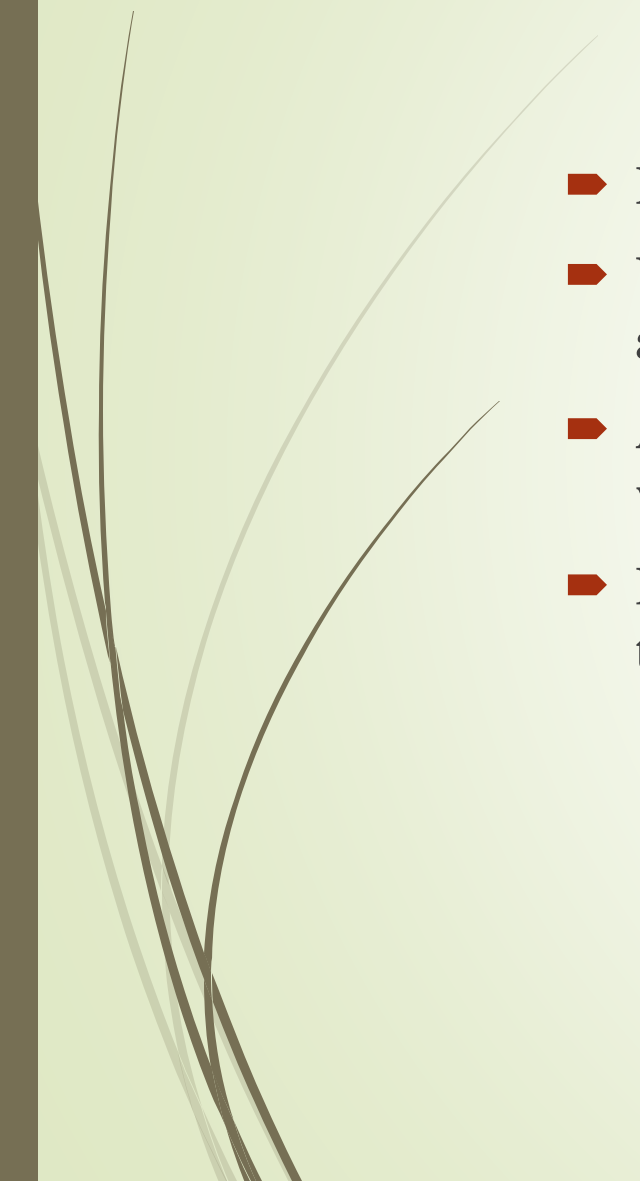
- Flask
- HTML, CSS, Bootstrap
- JavaScript, JQuery

# Flow Chart





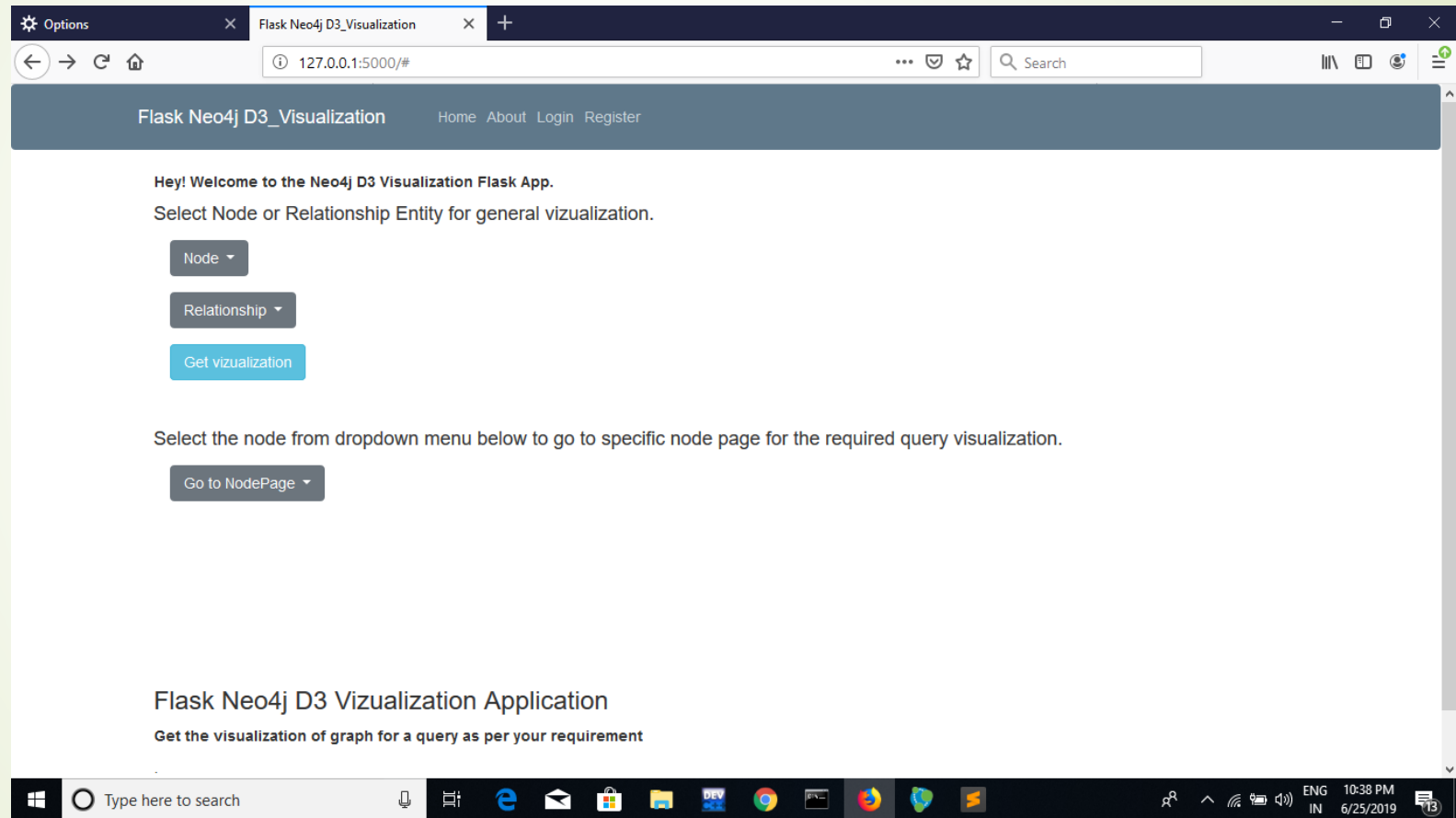
# Working of the Application

- In Neo4j, the knowledge graph is running for the datasets provided.
  - User will give the input as node name, relationship type, etc. and then accordingly the query will be fired.
  - After that when user will hit the visualize button, he will be directed to the visualization of the output of the fired query.
  - In the Visualization page, User can see the required information via the property on the top and color code provided below on the page.
- 

# Json Format which is taken as input for the data visualization of the fired query

```
{
  "results": [
    {
      "columns": ["DRUG", "DISEASE"],
      "data": [
        {
          "graph": {
            "nodes": [
              {
                "id": "1",
                "labels": ["DRUG"],
                "properties": {
                  "name": "drug1"
                }
              },
              {
                "id": "7",
                "labels": ["DISEASE"],
                "properties": {
                  "name": "disease1"
                }
              }
            ],
            "relationships": [
              {
                "id": "72",
                "type": "dr_ds",
                "startNode": "1",
                "endNode": "7",
                "properties": {
                  "name": "dr_ds1"
                }
              }
            ]
          }
        }
      ]
    }
  ],
  "errors": []
}
```

# User Interface



# User Interface

The screenshot displays a web browser window with the address bar showing '127.0.0.1:5000/drug'. The page title is 'Drug information search page'. The interface contains five distinct search sections, each with a text input field, a 'submit' button, a 'Reset' button, and a 'Visualize' button.

**Section 1:** Enter the drug node name and search for visualization of the node. Input: 'type drug name'.

**Section 2:** Enter the drug node name and then enter the entity name by selecting the entity from dropdown menu to get relation of that node with that particular entity. Inputs: 'type drug name' and 'type entity name'. Dropdown menu: 'Entity'.

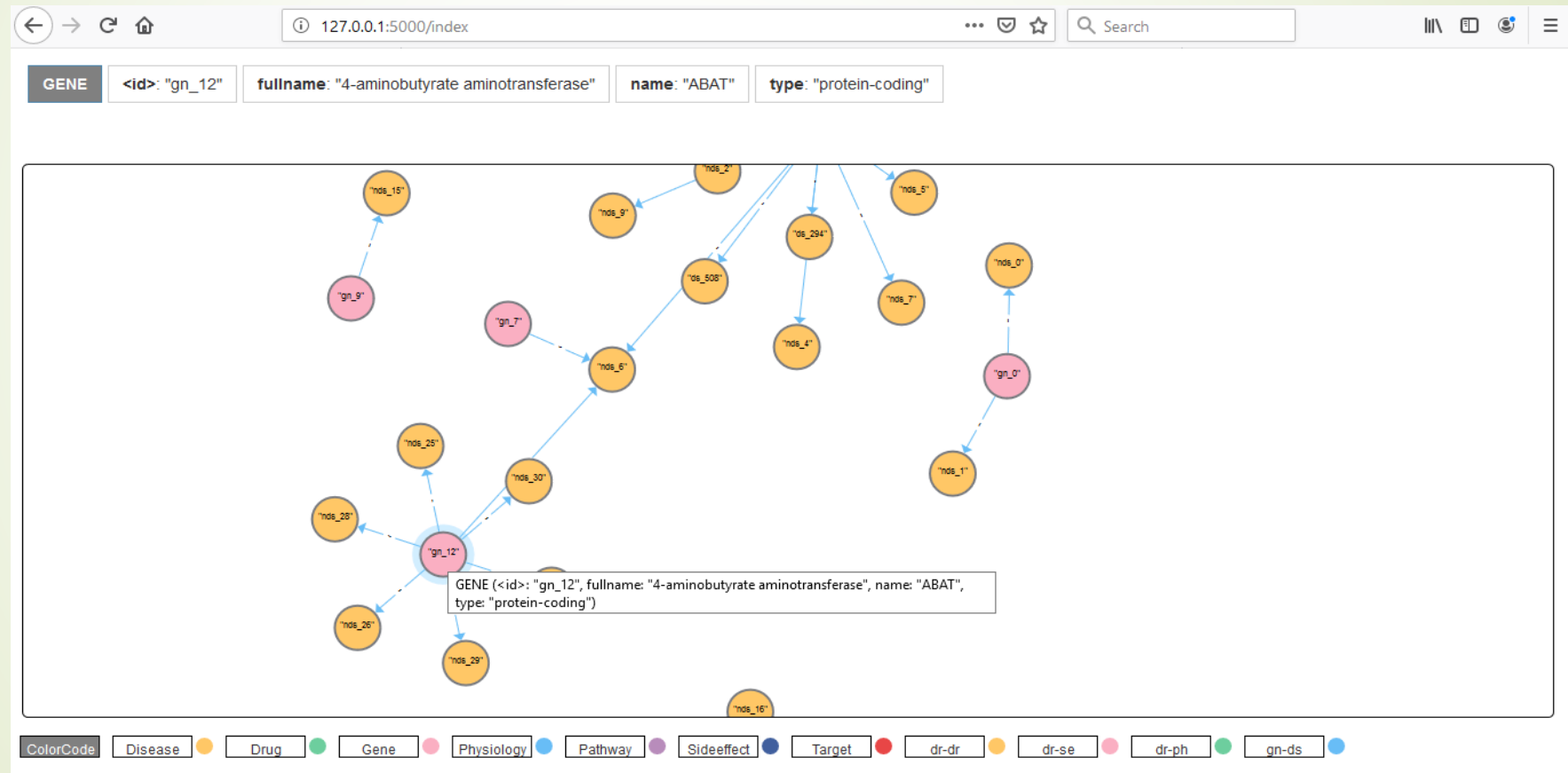
**Section 3:** Enter the drug node name and then select the entity to get all the relations between node and the selected entity. Inputs: 'type drug name' and 'Entity'.

**Section 4:** Enter the drug node name and then search all the connetions existing in shortest path of length L. Inputs: 'type drug name' and 'Length'.

**Section 5:** Search indegree or outdegree from the below radio buttons and then search the subgraph of selected degree of node. Radio buttons: 'Indegree' (selected) and 'Outdegree'. Input: 'Degree of node'.



# Visualization View





**Thank You!**