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

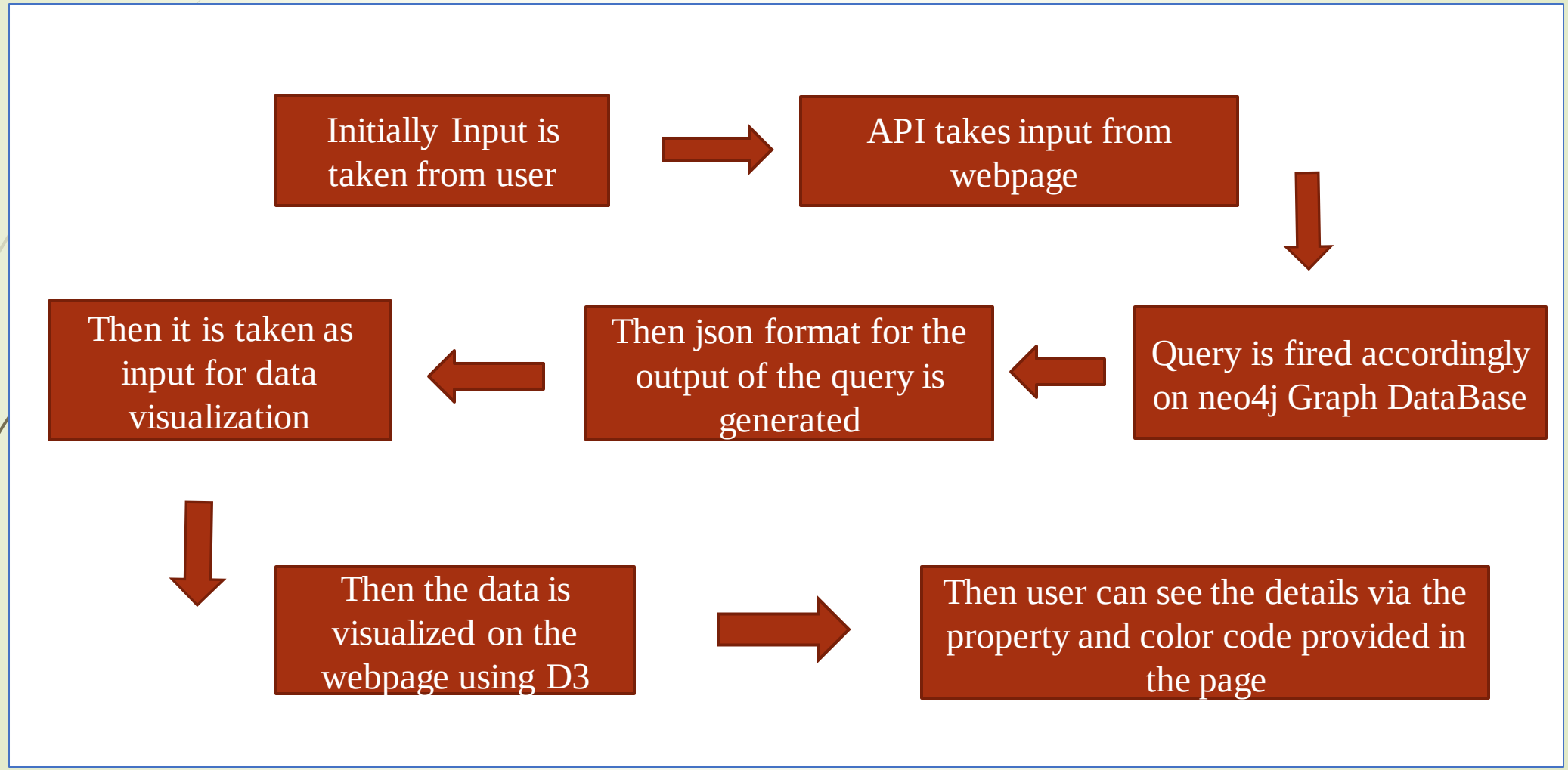
For Data Visualization

- Data Driven Documents (D3)
- Scalar Vector Graphics

For API and User Interface

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

Flow Chart


Initially Input is taken from user
API takes input from webpage
Query is fired accordingly on neo4j Graph DataBase
Then json format for the output of the query is generated
Then it is taken as input for data visualization
Then the data is visualized on the webpage using D3
Then user can see the details via the property and color code provided in the page

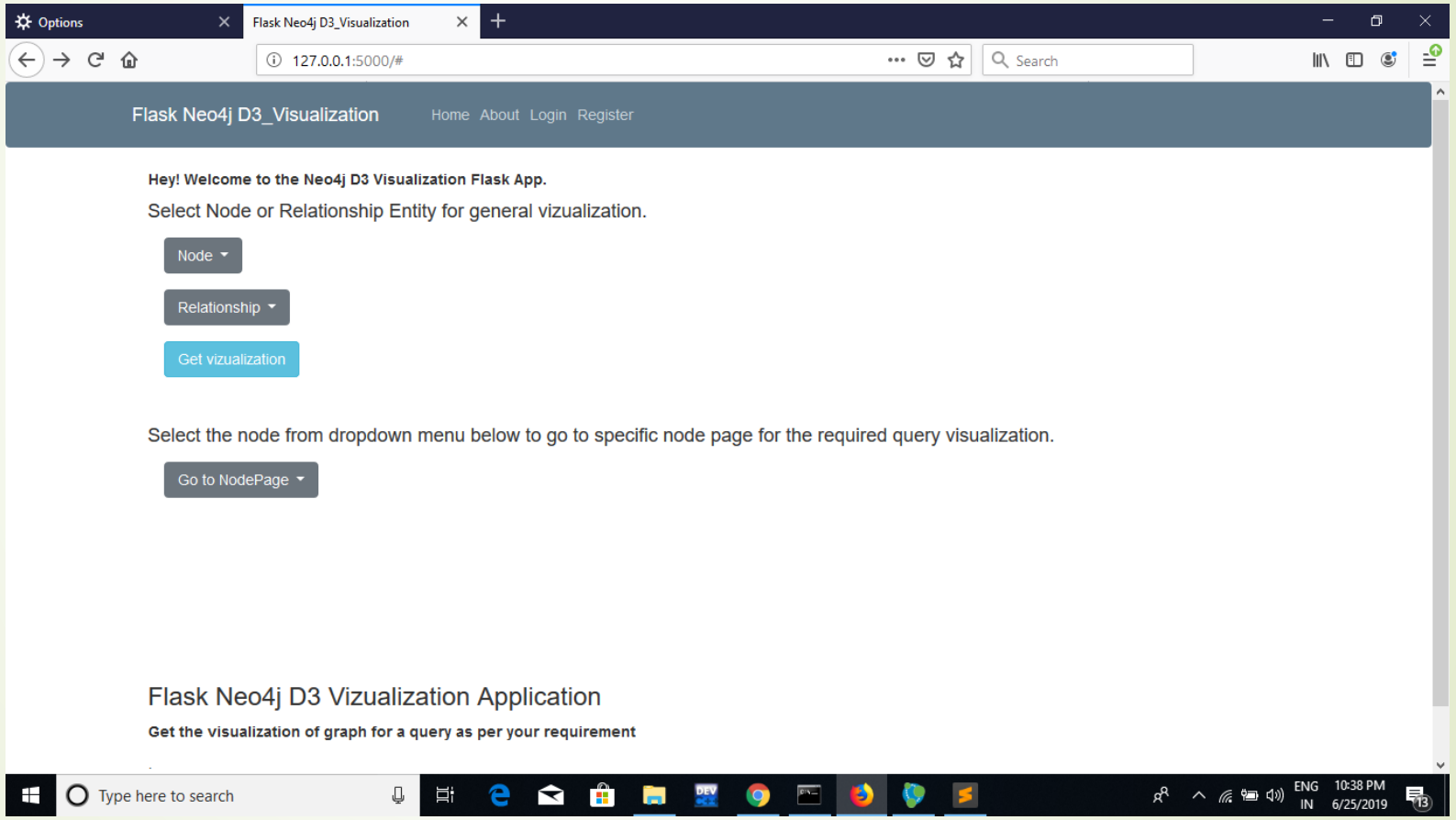
Working of the Application

- 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.

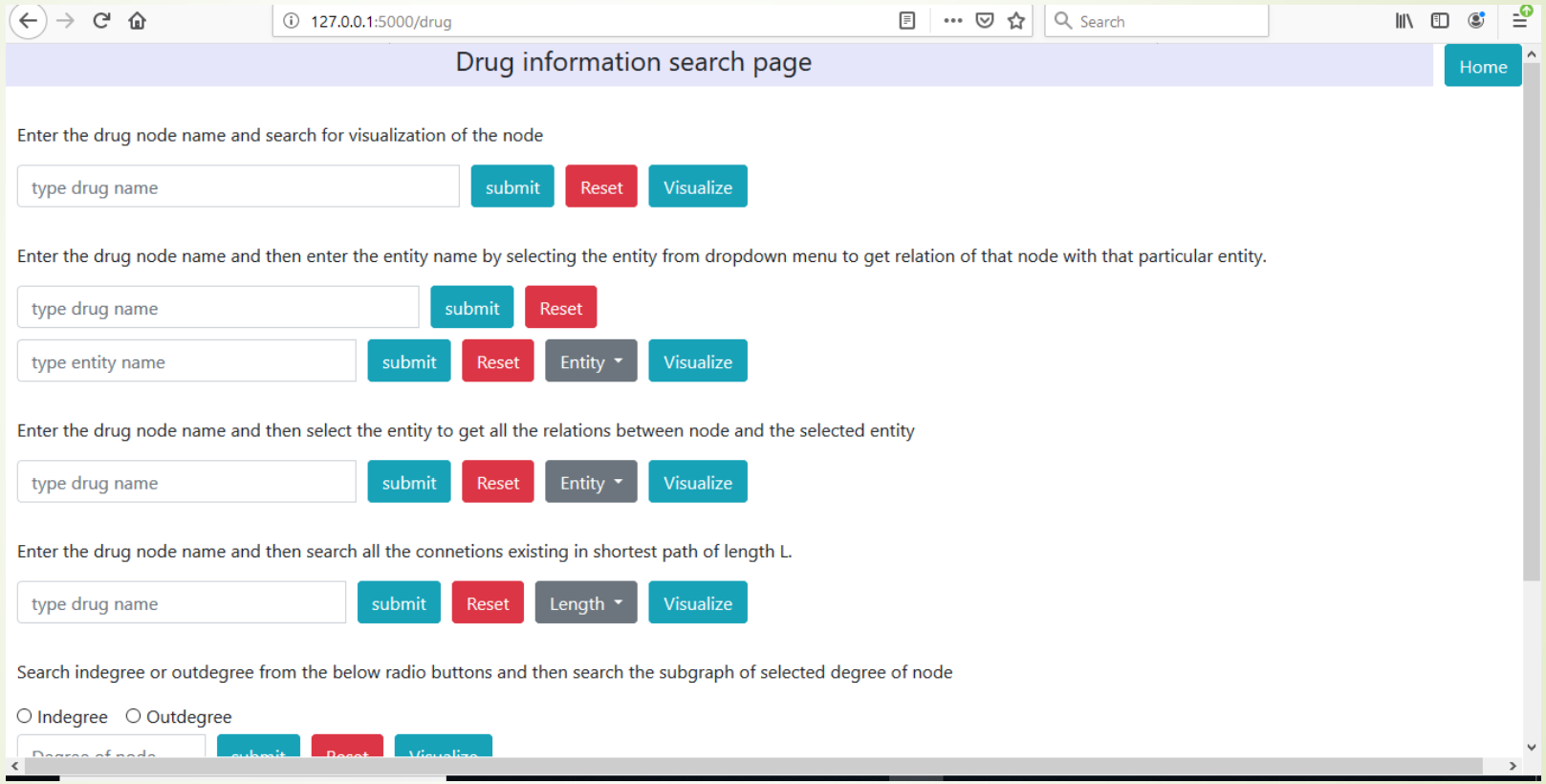
Json Format which is taken as input for the data visualization of the fired query


```
{
    "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": []
}
```

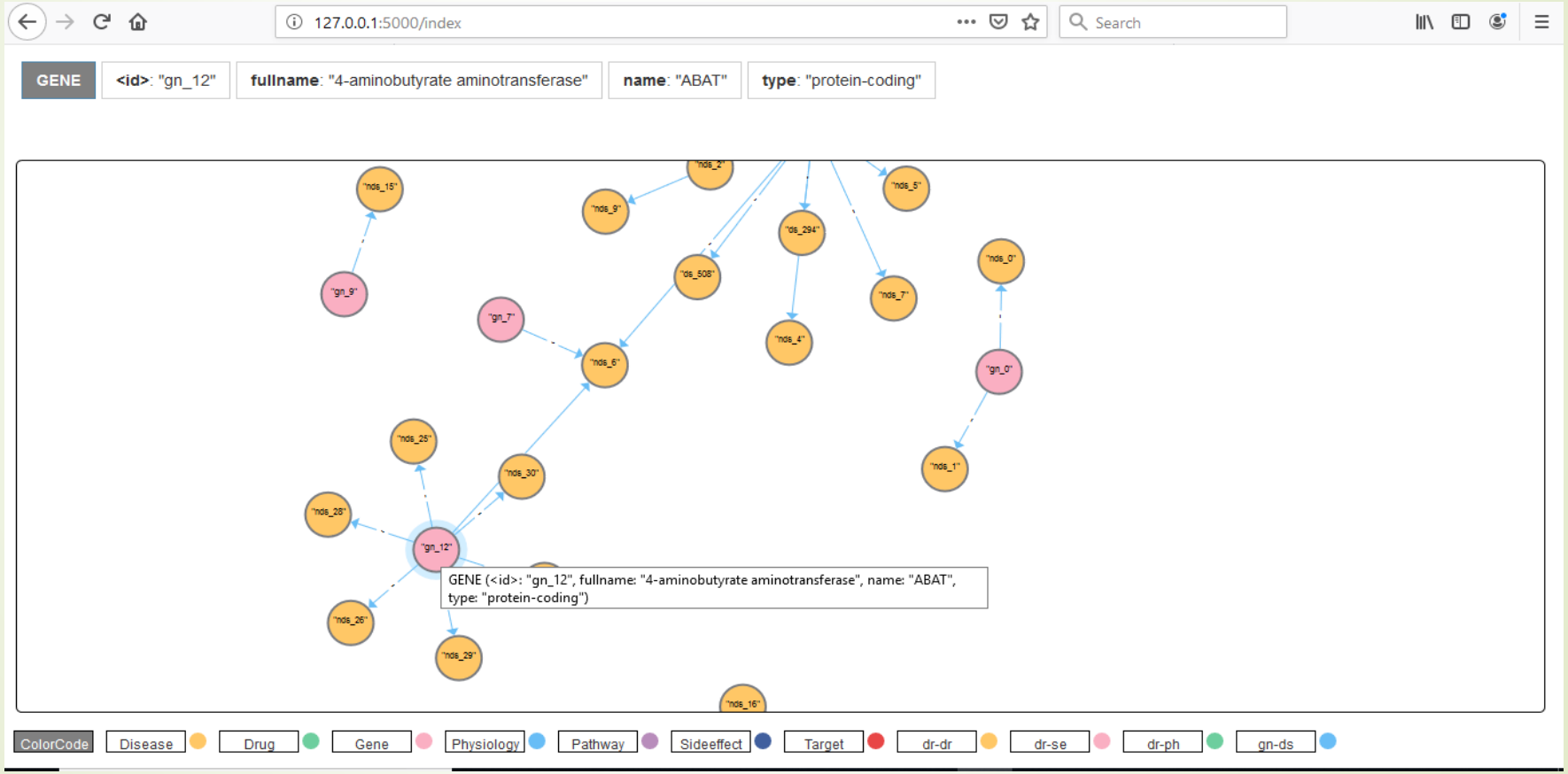
User Interface



User Interface



Visualization View



Thank You!