**Artificial Intelligence (AI)**

**(Assignment 01)**

**Submitted to:**

Zuhaib Hussain Butt

**Group Name & Section:**

Visionaries - B

**Group Members:**

|  |  |
| --- | --- |
| **Name** | **Roll No.** |
| Farwa Arshad | 211400157 |
| Muhammad Jamal | 211400019 |
| Saad Abid | 211400011 |
| Wajeeh-ur-Rehman | 211400021 |

**Introduction:**

Interactive Graph Visualization is a web application that allows you to create and visualize graphs on the fly and run various fundamental graph algorithms like BFS, DFS and UCS. The edge costs and source and destination nodes can be specified using a simple form, allowing users to add nodes and edges to the graph. The user selects an algorithm, inputs the start and goal nodes, and thereafter the relevant processing algorithm processes the graph to process the path and total cost which will be shown as an output. It enables to set up the graph, add edges in an ongoing way, visualize how the updates aligns with the canvas, calls search algorithms and reset the graph on memory to be reused. This helps you understand graph structures and direct traversal algorithms interactively.

**Technologies Used:**

* **HTML:** Used to framework the website structure some buttons and input forms.
* **CSS:** Used the CSS for the styling the website, to make it look better and usable.
* **JavaScript:** Used for writing the algorithms (BFS, DFS, and UCS), as well as for interactively updating the graph.

**Project Implementation:**

1. **Algorithm Implementation**
   * Finding paths the other way round, we implemented BFS: Breadth-First Search, keeping a queue to traverse the paths layer by layer.
   * DFS: Depth-First Search Data structure stack, Deepest node first
   * UCS: Uniform-Cost Search used a priority queue to explore paths in order of the lowest cost.
2. **Website Functionality**
   * The website has buttons to choose an algorithm (BFS, DFS, UCS) and inputs to add nodes and edges to the graph.
   * The path, total cost, and steps taken are displayed dynamically based on user input.

**Challenges Faced:**

* It was trying to revisit technologies after a long time which forced us to explore again and spend ample of time in getting comfortable with the tools.
* Adding and removing nodes and edges dynamically through the UI and rendering on the canvas was something we never tried before.
* Learning how to dynamically add and display nodes and edges was a key milestone in the project.
* We had read about traversal algorithms like BFS, DFS, UCS theoretically and visually, but coding this up required deeper understanding.
* After last place course, information security, where we were about the encryption part and the reversing of algorithm, we now understood the logic behind traversal algorithm.
* They designed and implemented the traversal algorithms with relative ease and efficiency due to their familiarity with algorithmic thinking.
* All these challenges led us to become much better technically as well as more confident working on more difficult features.

**Conclusion:**

The project helped us with the technical know-how and combined the theoretical concepts with practical implementation, specifically the graph traversal algorithms like BFS, DFS and UCS. We discovered how to dynamically create nodes or edges using a UI, producing interactive applications. The experience of getting accustomed with the existing knowledge about encryption algorithms sped up the process, while facing the problems reinforced our problem-solving and team working skills. In conclusion, this project was not only a great learning experience for our group but also enhanced our confidence in problem-solving with algorithms.