## **Bytexl’s Guided Project Overview**

**Title:** Interactive Graph Algorithms Visualizer  
**Duration:** Approximately 2 hours  
**Programming Language:** Python  
**Platform:** Nimbus on Bytexl

**Purpose:**This project offers students a practical, hands-on approach to learning essential graph algorithms such as Breadth-First Search (BFS), Depth-First Search (DFS), and Dijkstra’s algorithm. By developing a visualizer, students will gain valuable experience in coding, implementing, and understanding how these algorithms function in real-time. This is an excellent opportunity for students preparing for technical interviews or wanting to deepen their knowledge of data structures and algorithms (DSA) applicable to real-world scenarios.

## **Project-Based Learning Course Overview**

### **About the Project**

The **Interactive Graph Algorithms Visualizer** is designed for students who wish to strengthen their understanding of key graph algorithms by implementing a visual tool where they can add nodes, connect edges, and select different algorithms to visualize their operations in real-time. This project is especially useful for students preparing for technical coding assessments, interviews, or roles requiring strong DSA knowledge.

### **Prerequisites**

* **Programming Skills:** Basic knowledge of Python.
* **Data Structures Knowledge:** Understanding of fundamental graph concepts, BFS, DFS, and Dijkstra’s algorithm basics.

### **Learning Objectives**

By the end of this project, students will:

1. Develop a clear understanding of BFS, DFS, and Dijkstra’s algorithm by observing their execution step-by-step on a visual graph.
2. Strengthen practical DSA skills by building and debugging graph algorithms.
3. Gain confidence in using Python for visualizing algorithms, which is beneficial for interviews and real-world applications.

### **Key Skills to Practice**

* **Algorithm Implementation:** Coding BFS, DFS, and Dijkstra’s algorithm in Python.
* **Graph Data Structures:** Understanding the structure and application of nodes, edges, and graph traversals.
* **Visualization & Debugging:** Building a visual interface for step-by-step algorithm testing and debugging.

### **Learning Platform: Nimbus on Bytexl**

Students will complete this project on **Nimbus**, Bytexl’s cloud-based learning platform designed for interactive, hands-on projects. Nimbus is pre-configured with Python and other development tools, removing the need for complex setup and allowing students to focus directly on coding and visualization.

**Platform Benefits:**

* **Python Environment:** A Python-based environment tailored for data structure and algorithm development.
* **Step-by-Step Instructor Guidance:** Supportive guidance at each step to ensure smooth project completion.
* **Development-Ready Tools:** Tools are pre-installed, allowing students to start coding immediately without setup.
* **Real-Time Interaction:** Execute, test, and visualize graph algorithms in real time on any desktop or laptop.

## **Step-by-Step Learning Guide**

The **Interactive Graph Algorithms Visualizer** is structured to be completed in approximately 2 hours, with specific tasks for clear learning progression.

### **Project Structure**

* **Task 1:** Set up the graph visualizer interface using Nimbus tools.
* **Task 2:** Implement BFS and DFS algorithms, integrating step-by-step visualization.
* **Task 3:** Add Dijkstra’s algorithm, showing real-time shortest path calculation.
* **Task 4:** Test and refine the visualizer, ensuring usability and accuracy in the visual display.

## **Educator’s Introduction**

Hi! I’m *Suraj Mourya*, your instructor for this course. I have a background in computer science engineering with over 3 years of teaching experience specializing in Data Structures and Algorithms. I’ve taught at more than 10 colleges across India and am certified in DSA, receiving consistent positive feedback from students. My goal is to build interactive, engaging learning experiences to support your growth in technical skills. When I’m not teaching, I enjoy diving into advanced topics in computer science and expanding my technical knowledge.

## **Completion and Certification**

Upon successfully completing the **Interactive Graph Algorithms Visualizer** project, students will have the opportunity to take a quiz to reinforce their learning. Scoring 80% or higher will earn them a completion certificate, validating their project achievement and DSA skills in Python. This certification will enhance their portfolio, making them better prepared for technical roles and coding challenges.