**Knapsack Optimization**

**Implementation Plan**

**Project Code**: \_\_\_\_\_\_\_\_\_

**Project Manager**: Sir Fahad Maqbool

**Project Advisor**: Ferzoq Khan

**Project Team**:

Fatima Naveed (BSSE51F20R002)

Saman Waheed (BSSE51F20R003)

Humna Ramzan (BSSE51F20R023)

**Submission Date:** 5th October 2023



|  |  |
| --- | --- |
| Week 1 | Research and understand the concept of Knapsack Optimization.  Define the project scope, objectives, and expected outcomes.  Create a detailed project plan, including milestones and deadlines**.** |
| Week 2 | Set up the development environment with JavaScript.  Choose a code editor and version control system (e.g., VS Code and Git).  Create a new project repository on GitHub. |
| Week 3 | Dive deeper into the Knapsack problem and its variations. Implement basic algorithms on paper to solve small instances of the problem. |
| Week 4 | Write JavaScript functions to represent items and the knapsack.  Implement a brute-force algorithm to solve the 0/1 Knapsack problem. |
| Week 5 | Study the dynamic programming approach for solving the Knapsack problem.  Implement a dynamic programming solution for the 0/1 Knapsack problem in JavaScript. |
| Week 6 | Test the dynamic programming solution with various scenarios.  Optimize the code and make it more efficient. |
| Week 7 | To Learn about the Fractional Knapsack problem and its greedy algorithm.  Implement the Fractional Knapsack algorithm in JavaScript. |
| Week 8 | Test the Fractional Knapsack algorithm with different item weights and values.  Document the code and provide clear explanations. |
| Week 9 | Implement additional features such as user input and item selection.  Create a user-friendly interface using HTML and CSS. |
| Week 10 | Add error handling to the application.  Test the entire application thoroughly. |
| Week 11 | Explore variations of the Knapsack problem, such as multiple knapsacks or bounded knapsacks.  Implement solutions for these variations in JavaScript. |
| Week 12 | Test the solutions for Knapsack problem variations.  Compare the performance of different algorithms for each variation. |
| Week 13 | Optimize the code for efficiency and readability.  Conduct code reviews or seek peer feedback for improvements. |
| Week 14 | Prepare comprehensive documentation, including a user guide and developer guide.  Finalize the project and perform a final round of testing. |
| Week 15 | Prepare a presentation outlining the problem, the solution, and the lessons learned during the project.  Demonstrate the project's functionality to the teacher. |
| Week 16 | Submit the project along with all documentation, code, and presentation materials.  Reflect on the project's challenges, achievements, and areas for future improvement in a project report. |