**DAYANANDA SAGAR UNIVERSITY**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**SCHOOL OF ENGINEERING**

**DAYANANDA SAGAR UNIVERSITY**

**KUDLU GATE**

**BANGALORE – 560068**

****

**MINI PROJECT REPORT**

***ON***

**“KNAPSACK TO THE RESCUE”**

**SUBMITTED TO THE 4TH SEMESTER ANALYSIS AND DESIGN OF ALGORITHMS-2020**

**BACHELOR OF TECHNOLOGY**

***IN***

**COMPUTER SCIENCE & ENGINEERING**

***Submitted by***

SPOORTHI KULKARNI - (ENG18CS0280)

SRISHTI RANJAN - (ENG18CS0284)

TEJASWINI . R - (ENG18CS0298)

 V.A.S KIRANMAYEE - (ENG18CS0304)

***Under the supervision of***

**Prof. Kavyashree**

**ACKNOWLEDGEMENT**

The satisfaction that accompanies the successful completion of task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts with success.

We are especially thankful to our **Chairman, Dr. M K Banga**, for providing necessary departmental facilities, moral support and encouragement.

We are very much thankful to **Prof. Kavyashree**, for providing help and suggestions in completion of this mini project successfully.

We have received a great deal of guidance and co-operation from our friends and we wish to thank all that have directly or indirectly helped us in the successful completion of this project work.

                    SPOORTHI V KULKARNI (ENG18CS0280)

SRISHTI RANJAN-(ENG18CS0284)

TEJASWINI REDDY-(ENG18CS0298)

V.A.S KIRANMAYEE-(ENG18CS0304)

**DECLARATION**

We hereby declare that the work presented in this mini project entitled-

“KNAPSACK TO THE RESCUE”, has been carried out by us and it has not been submitted for the award of any degree, diploma or the mini project of any other college or university.

SPOORTHI V KULKARNI (ENG18CS0280)

SRISHTI RANJAN-(ENG18CS0284)

TEJASWINI REDDY-(ENG18CS0298)

V.A.S KIRANMAYEE-(ENG18CS0304)

**ABSTRACT**

While travelling there is often a dilemma where you have to consider the cabin baggage weight limit at the airport and according to that you have to choose between different items you want. Deciding what to leave behind entails laying out all my things and choosing which ones to keep. That decision is based on the item’s usefulness, worth and its weight, decisions like this could be made more efficiently by a computer. It' s done so frequently in fact, that many will recognize this scenario as the classic packing problem or knapsack problem. Hence we solve the knapsack problem using the Greedy method. A greedy algorithm is a one-pass algorithm that constructs a single final solution. At each stage of the problem, the greedy algorithm picks the option that is locally optimal, it looks like the most suitable option right now. There are three main operations we need to do, Sort items by worth and weight. Put an item in the bag. Check to see if the bag is full.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **SNO** | **TOPICS** | **PAGE NO.** |
| 1. | Introduction | 1 |
| 2. | Literature Review | 2 |
| 3. | Problem Statement | 3 |
| 4. | System Design  4.1 System design  4.2 Test case | 4-8 |
| 5. | Expected advantages and limitation | 9 |
| 6. | Conclusion and future enhancements | 10 |

**TABLE OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **SNO** | **FIGURES** | **PAGE NO.** |
| 1 | Fig 4.1.1 Step 1 code snippet | 4 |
| 2 | Fig 4.1.2 Step 2 code snippet | 5 |
| 3 | Fig 4.1.3 Step 3 code snippet | 5 |
| 4 | Fig 4.1.4 Step 4 code snippet | 6 |
| 5 | Fig 4.1.5 A flowchart of the basic working of the algorithm | 6 |
| 6 | Fig 4.2.1 Output of Test case 1 | 7 |
| 7 | Fig 4.2.2 Output of Test case 2 | 7 |
| 8 | Fig 4.2.3 Output of Test case 3 | 8 |
| 9 | Fig 4.2.4 Output of Test case 4 | 8 |