**THE ISLAMIA UNIVERSITY OF BAHAWALPUR**

**TODOLIST**

**SUBMITTED TO:**

**Sir Usman Ghani**

**SUBMITTED BY:**

** Muhammad Talha**

**ROLL NUMBER:**

**S23BDOCS1M01121**

**SEMESTER:**

**3rd**

**SECTION:**

**BSCS – 2M**

Project Report: To-Do-List Application

**1. Introduction**

The ToDo List Application is a robust task management tool designed to streamline the process of organizing, tracking, and prioritizing tasks. This report presents an indepth analysis of the project, covering its objectives, design principles, implementation details, user interface, and potential areas for future enhancement.

**2. Objectives**

The primary goal of the ToDo List Application is to provide users with a flexible and intuitive platform for managing their tasks effectively. Key objectives include:

* Facilitating seamless task addition, removal, and status updates.
* Offering versatile task retrieval options based on priority, status, and category.
* Providing insightful statistics to help users gain better insights into their task management habits.

**3. Design Overview**

**3.1. Class Structure**

The project is structured around three main classes:

Task: Represents an individual task, encapsulating essential attributes and behaviors.

Node: Serves as a building block for the linked list data structure, facilitating efficient task storage and traversal.

ToDoList: Acts as the orchestrator, managing the task list and exposing a userfriendly interface for interaction.

**3.2. Data Structures**

The project leverages a doubly linked list data structure for task management, offering efficient insertion, deletion, and traversal operations.

**4. Implementation Details**

**4.1. Task Class**

The Task class encapsulates taskspecific attributes such as ID, description, priority, category, and status. Additionally, it provides methods for printing task details and updating task status.

**4.2. Node Class**

The Node class represents a node within the linked list, housing a Task object and pointers to the next and previous nodes. This enables seamless navigation through the task list.

**4.3. ToDoList Class**

The ToDoList class serves as the core component of the application, offering functionalities for task manipulation, retrieval, and statistical analysis. It provides a cohesive interface for users to interact with their tasks efficiently.

**4.4. Main Program**

The main program encapsulates the user interface logic, presenting users with a menudriven interface for performing various tasks such as adding, removing, updating, and viewing tasks.

**5. Usage Instructions**

**5.1. Adding a Task**

Choose option 1 from the menu.

Enter task details such as ID, description, priority, and category.

**5.2. Removing a Task**

Choose option 2 from the menu.

Enter the ID of the task to be removed.

**5.3. Updating Task Status**

Choose option 4 from the menu.

Enter the ID of the task whose status you want to update.

Choose the new status from the menu.

**5.4. Viewing Tasks**

Choose options 3, 5, 6, or 7 from the menu to view tasks based on priority, status, or category.

**6. Conclusion**

In conclusion, the ToDo List Application offers a comprehensive solution for efficient task management. By providing users with intuitive functionalities and insightful analytics, it empowers individuals to stay organized, focused, and productive in their daily endeavors.

**7. GitHub Repository**

The source code for the To-Do List Application is available on GitHub for collaboration, contributions, and further exploration. You can access the repository at the following link:

Feel free to clone the repository, submit bug reports, suggest improvements, or contribute to the project's development.