**COLLEGE OF BUSINESS EDUCATION**

****

**DODOMA CAMPUS**

*Student’s Name:* **ABAS CHIMSALA**

Reg no: **03.4631.01.02.2023**

***Course* : BIT**

***Lecturer*:madam ATUPELE CAIRO MWAITETE**

***Subject:* PROGRAMMING IN JAVA**

***Nature of Work:* INDIVIDUAL ASSIGNMENT**

**Question.**

You are required to create a small Java application that addresses an everyday challenge faced by individuals or communities in Tanzania with a theme of **"Digital Solutions for Everyday Challenges in Tanzania"**. Each student should select a specific challenge and provide a software-based solution.

**Report: Digital Study Organizer**

**Introduction**

The Digital Study Organizer project is a Java-based desktop application designed to help students manage their study schedules, stay organized, and boost productivity. Through the use of Java Swing, I built a user-friendly interface that integrates task management, scheduling, progress tracking, and a Pomodoro timer. The idea behind this project was to create a tool that not only helps learners stay on top of their tasks but also motivates them to stay productive. Below, I’ll provide a detailed explanation of the key features, some screenshots of the interface, and the challenges I faced during development.

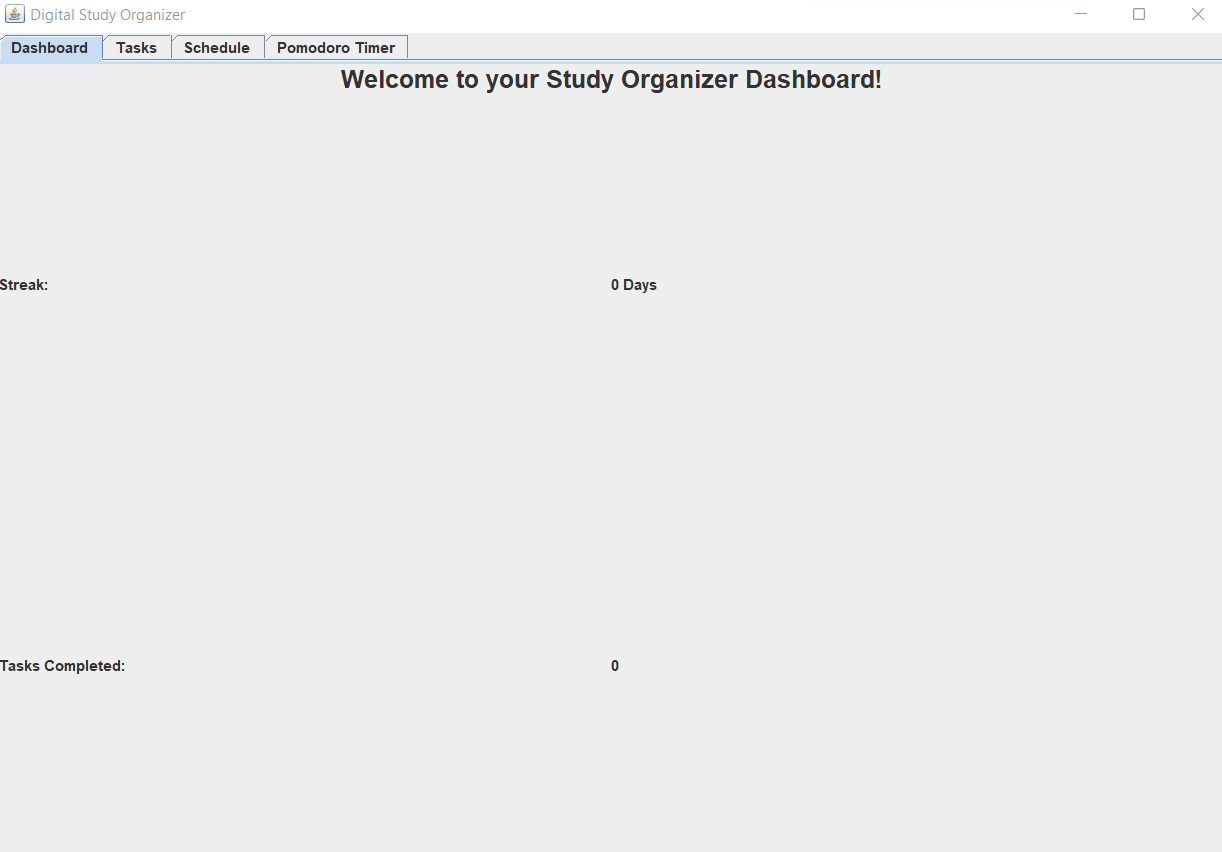
**Features Implemented**

1. **Dashboard**:
   * The dashboard is the first thing users see when they open the application. It serves as an overview of their current productivity and task progress.
   * It shows:
     + **Tasks completed**: A simple counter that tracks how many tasks the user has marked as "completed."
     + **Current streak**: This tracks the number of consecutive days that the user has completed at least one task. It’s a motivational metric, helping users stay consistent with their study habits.
   * The dashboard’s layout is simple but effective, designed to give users an instant overview of their progress without overwhelming them with too much information.
2. **Task Management**:
   * This feature allows users to create, update, and track tasks. Each task can have:
     + **Title**: The name of the task.
     + **Description**: A brief explanation of the task.
     + **Category**: To classify tasks (e.g., Study, Work, Personal).
     + **Priority**: Users can set the task’s priority (High, Medium, Low).
     + **Deadline**: Users can specify a deadline for each task.
     + **Status**: Tasks can be updated through three statuses: To-Do, In Progress, and Completed.
   * A table displays all tasks at a glance, and users can filter or sort tasks by priority, status, or deadline. This helps them stay organized and focused on what’s most important.
3. **Scheduling**:
   * The scheduling feature lets users input their weekly study or work schedule into a text area. This is particularly helpful for learners who want to visualize their week and allocate study blocks effectively.
   * The **Save Schedule** button ensures that the schedule is preserved, so users don’t have to enter it repeatedly.
4. **Pomodoro Timer**:
   * The Pomodoro Technique is a productivity method that encourages users to work in short, focused intervals (typically 25 minutes) followed by a short break.
   * This feature includes:
     + A timer that starts at **25:00** minutes, with visual feedback that counts down in real-time.
     + **Alerts** to notify the user when a Pomodoro session is completed.
     + When a session ends, the streak count updates, so users can track their progress over time.
   * The Pomodoro timer has been proven to help users maintain focus and avoid burnout.
5. **Progress Tracking**:
   * The app tracks and displays how many tasks have been completed and how many days in a row tasks or Pomodoro sessions have been finished.
   * This feature helps users stay motivated by showing the progress they’re making. The progress is updated automatically when tasks are marked as complete or Pomodoro sessions finish.

**Screenshots of the Project Interface**

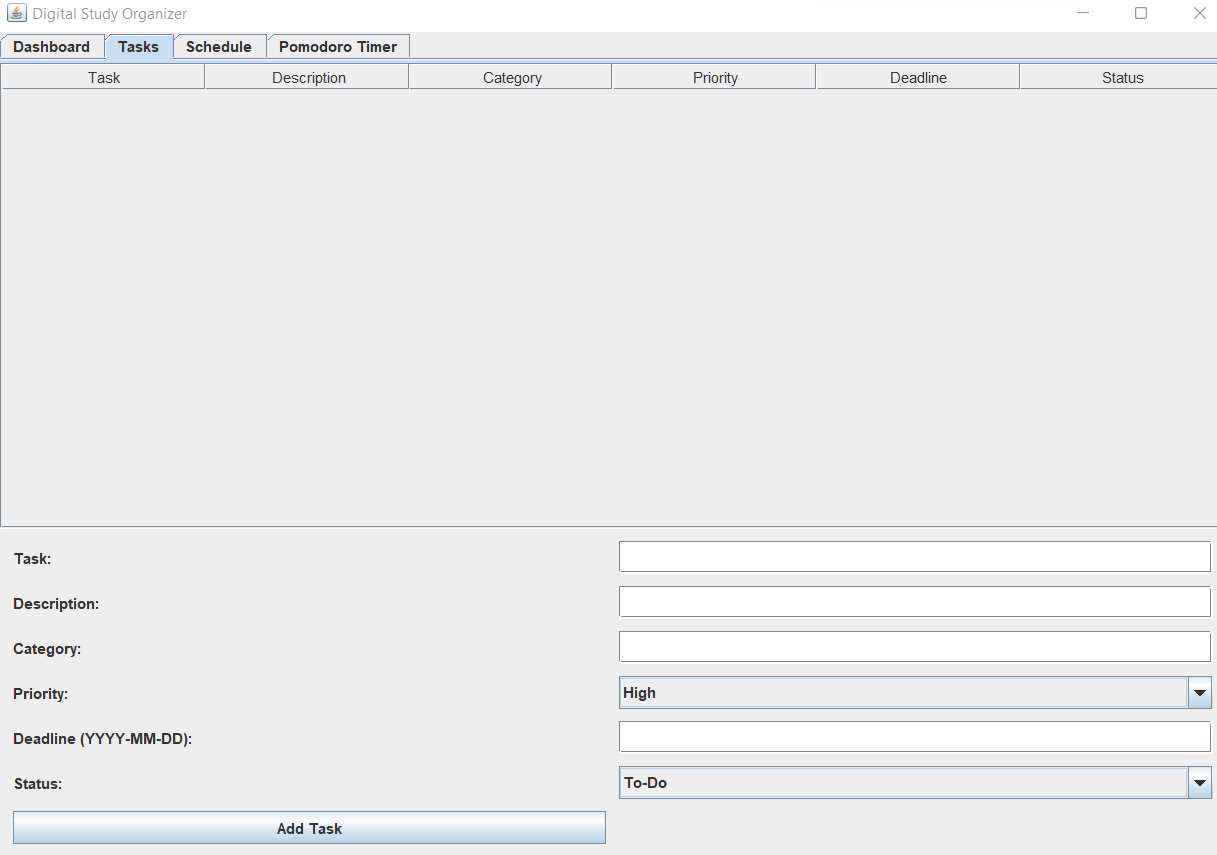
**1. Dashboard Panel**:

* A clean, simple design that displays overall metrics at a glance.
* Key stats such as tasks completed and current streak are clearly shown, offering quick motivation.



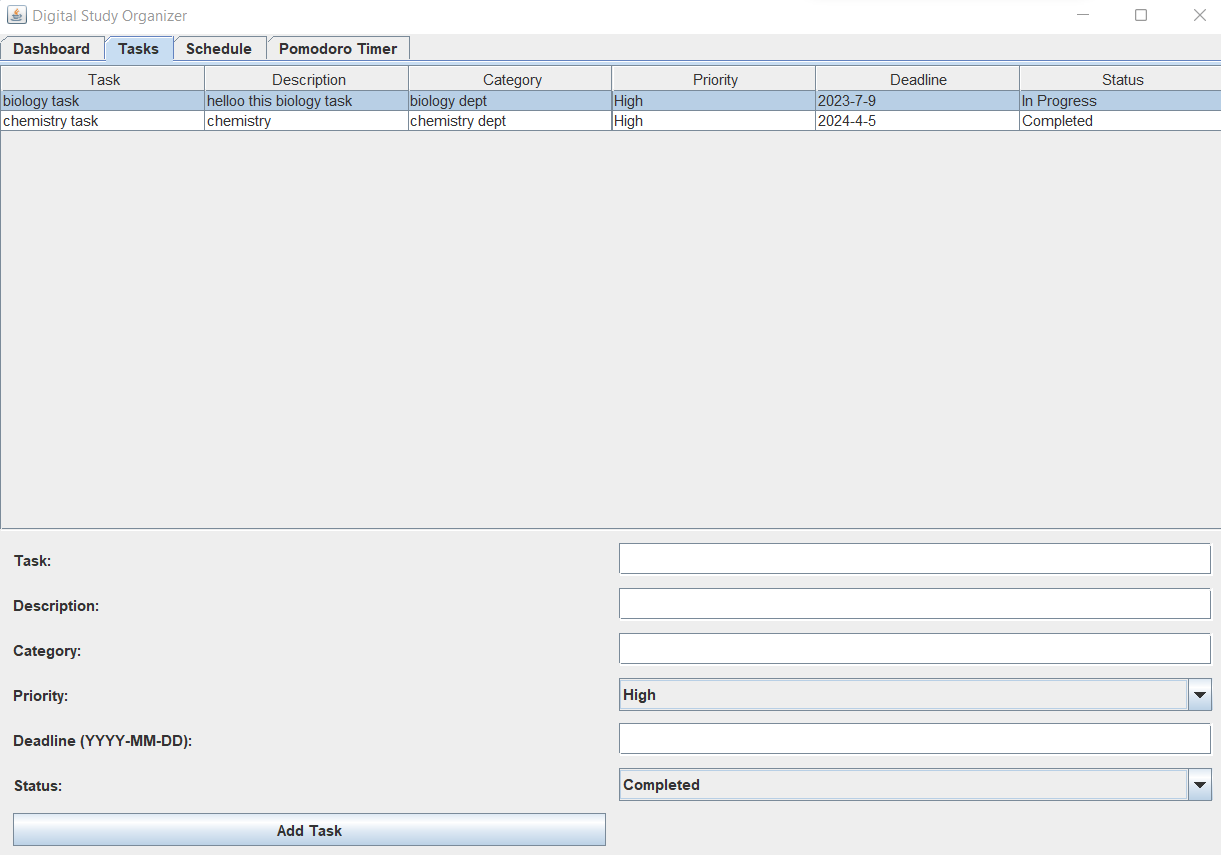
**2. Task Management Panel**:

* A table that organizes tasks by their title, priority, deadline, and status.
* Users can easily add or update tasks through input fields and buttons.



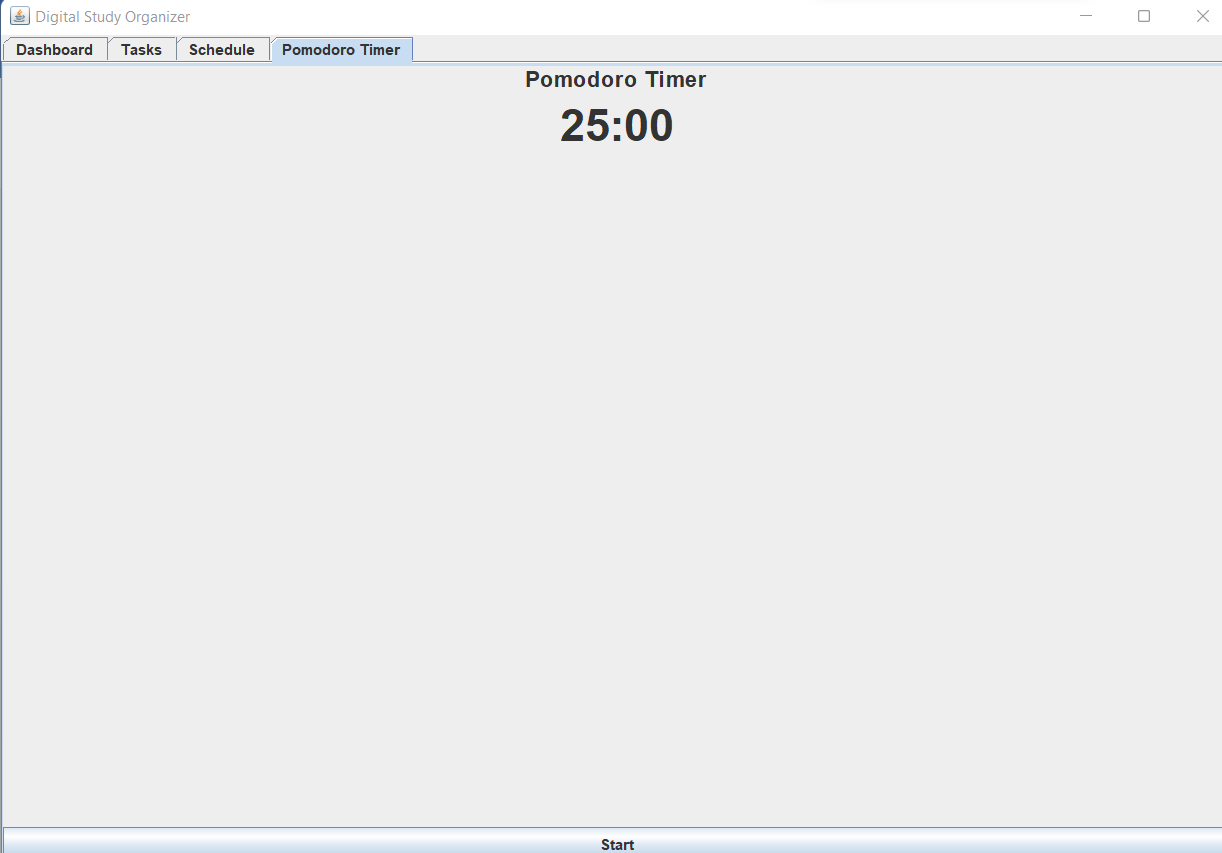
**3. Schedule Panel**:

* This panel features a text area where users can enter their weekly schedule.
* It’s simple yet effective for helping learners stay on track with their weekly planning.



**4. Pomodoro Timer Panel**:

* Displays a large countdown timer that starts at 25 minutes.
* The interface includes a “Start” button to begin the timer and visual feedback to track time.



**Challenges Faced During Development**

1. **GUI Design**:
   * One of the biggest challenges was designing an interface that was both functional and easy to use. Java Swing provided a variety of components, but I had to carefully select and arrange them to create a layout that felt intuitive.
   * It was difficult to strike a balance between providing enough functionality and keeping the interface clean and simple.
2. **State Synchronization**:
   * As the user interacts with different parts of the application (e.g., task management and scheduling), I needed to ensure that changes were reflected across all panels in real time.
   * For example, when a task is marked as complete, the dashboard’s task count and streak needed to update immediately. Implementing this feature required managing data models carefully and using event listeners for real-time synchronization.
3. **Progress Tracking Logic**:
   * The logic for tracking task completion and streaks was tricky. I had to develop algorithms that dynamically calculated these values every time a task was completed or a Pomodoro session finished.
   * Ensuring accuracy required extensive testing and debugging, as I needed to account for edge cases like users skipping sessions or tasks being edited.
4. **Timer Implementation**:
   * The Pomodoro timer’s implementation involved using multi-threading. I used the javax.swing.Timer class to create the countdown timer, but performance tuning was necessary to avoid delays or freezes during long sessions.
   * Handling the timer’s start, pause, and reset actions also required careful synchronization to prevent issues where the timer could malfunction.
5. **Error Handling**:
   * Throughout development, I had to ensure that the app handled incorrect inputs gracefully. For instance, users could mistakenly leave task descriptions blank or enter invalid dates, and I needed to display appropriate error messages.
   * I also had to ensure that user inputs followed proper formats, such as task deadlines being in a valid date format.
6. **Testing and Debugging**:
   * The development process included a lot of testing, especially with features like the Pomodoro timer and task management. For example, I encountered issues where the "Start" button could be clicked again while the timer was running, which led to multiple sessions starting.
   * Regular testing and debugging helped resolve these minor bugs and ensured the application worked as expected.

**Conclusion**

The Digital Study Organizer project successfully integrates essential features for students and learners who want to stay organized and productive. From task management to Pomodoro timers and progress tracking, the app offers a well-rounded solution for boosting productivity. Despite facing challenges during the development process, such as designing the interface and implementing real-time synchronization, I’m proud of the result. This tool is a valuable example of how digital solutions can support learning and personal organization.

**Future Improvements**

1. **Cloud Syncing**:
   * Implementing cloud syncing would allow users to access their data from multiple devices, making the app more versatile.
2. **Collaboration Tools**:
   * Introducing features for group projects, like shared task boards, could make the app more suitable for team-based work.
3. **Enhanced Analytics**:
   * I’d like to add more detailed performance insights, such as task completion trends by category or subject.
4. **Customization Options**:
   * Giving users more control over the app’s theme and layout would enhance the user experience, allowing for personalization.