**Final Report:** Online Unused Medicine Donation System

**a) Project Introduction**

The Online Unused Medicine Donation System is a web-based application that allows individuals to donate unused medicines, and others in need to request and receive them. The platform also supports feedback submission, admin approval for donations, and login-based security for all users. This system helps reduce medicine waste while supporting healthcare for underprivileged communities.

**Technologies Used:**

* Frontend: Windows Forms (C#)
* Backend: C# with Windows Forms logic
* Database: Text files

**b) User Stories (All Sprints)**

**Sprint 1:**

1. As a user, I want to register and log in so that I can access the platform.
2. As a user, I want to donate unused medicines by filling out a form.
3. As an admin, I want to view all registered users.

**Sprint 2:**

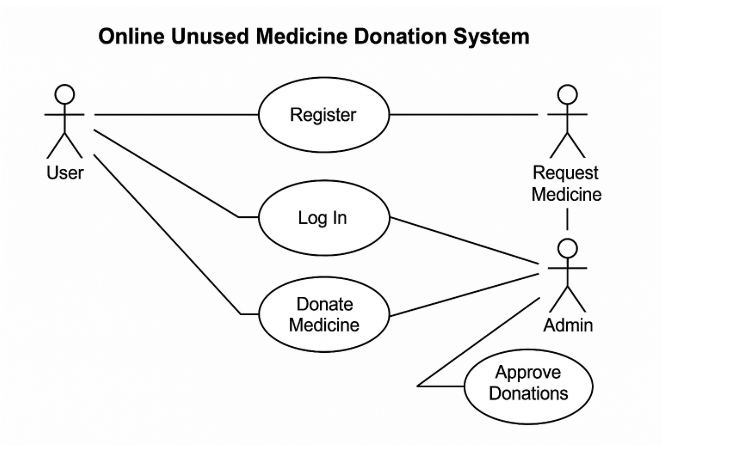
1. As a user, I want to search for medicines using filters so that I can find what I need.
2. As a user, I want to request medicines from the available stock.
3. As an admin, I want to manage medicine stocks and approve requests.

**Sprint 3:**

1. As a user, I want to submit feedback after using the system so that I can share my experience.
   * Sub-stories:
     + Submit feedback form after donation or request.
     + View previous feedback (if allowed).
2. As a system user, I want to be required to log in before accessing any feature so that my data is secure.
   * Sub-stories:
     + Redirect to login if not authenticated.
     + Prevent unauthorized access to dashboard or forms.
3. As an admin, I want to approve medicine donations before they are listed so that only valid medicines are available.
   * Sub-stories:
     + Admin reviews each new donation.
     + Admin accepts/rejects with reason.

**c) Design (Sprint 2 Items)**

**1. Use Case Diagram**



**2. Data Flow Diagram (DFD)**

**Level 0:**

1. External entities: User, Admin
2. Processes: Register/Login, Donate Medicine, Request Medicine, Approve Donation
3. Data Stores: Users txt, Medicine txt, Orders txt, Feedback txt.

**Level 1:**

1. **Login/Registration**: Inputs credentials, returns authentication
2. **Medicine Donation**: Inputs medicine info → stored as pending
3. **Admin Approval**: Admin reviews and sets approval status
4. **Medicine Request**: Request form → adds to orders txt, updates stock

**d) Architecture**

**Architecture Pattern:** Three-Tier Architecture

#### 1. **Presentation Layer – View (Windows Forms)**

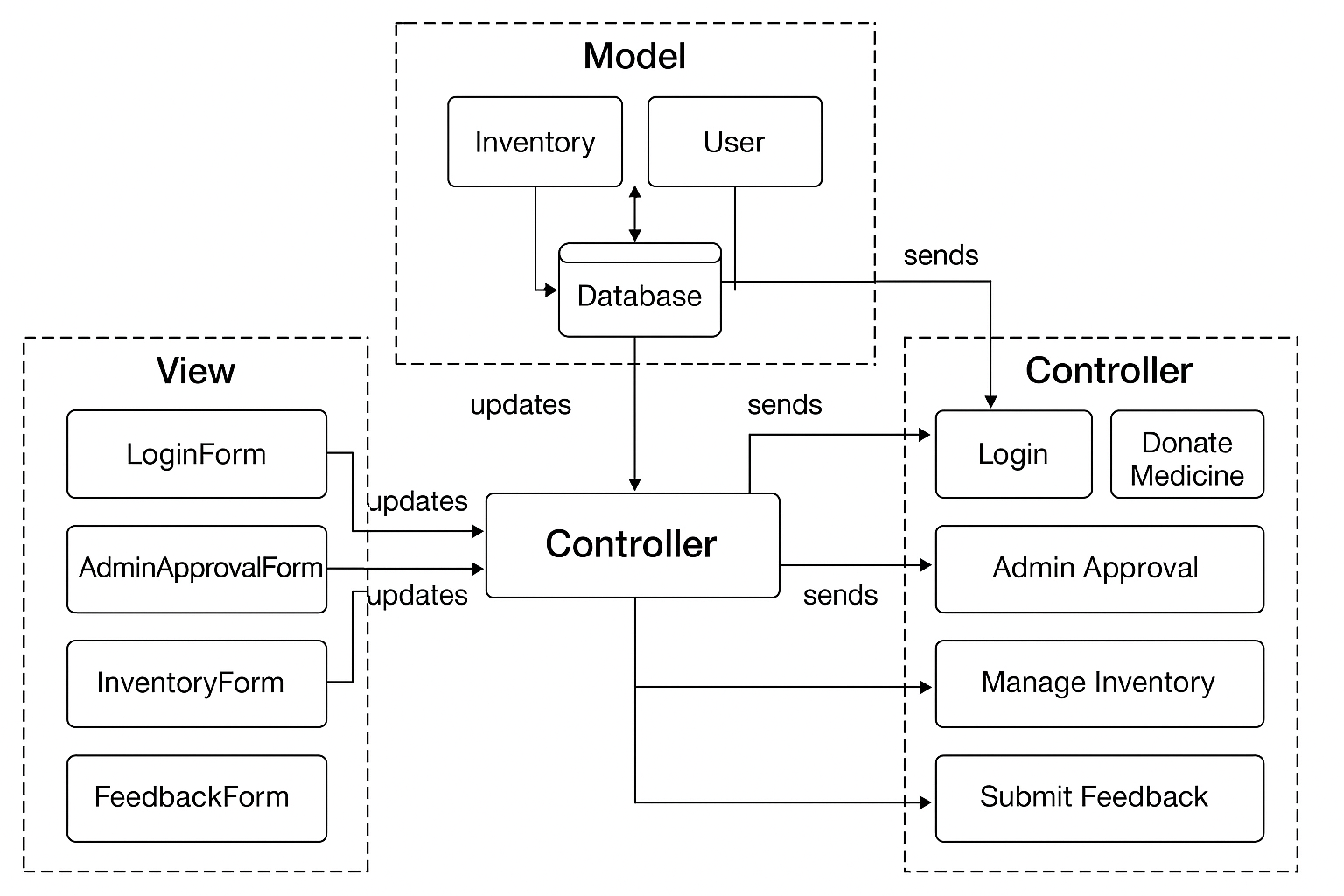
* Implements the **UI** using WinForms (e.g., LoginForm, RegisterForm, DonateForm).
* Interacts directly with users to capture input and display output.
* Each form represents a **View** in the MVC model.

#### 2. **Application Layer – Controller (Code-Behind Logic)**

* Contains the **business logic** and serves as the **Controller** in MVC.
* Handles actions like:
  + Button clicks
  + Input validation
  + Workflows (e.g., admin approval, sending notifications)
* Acts as the bridge between View (UI) and Model (data).

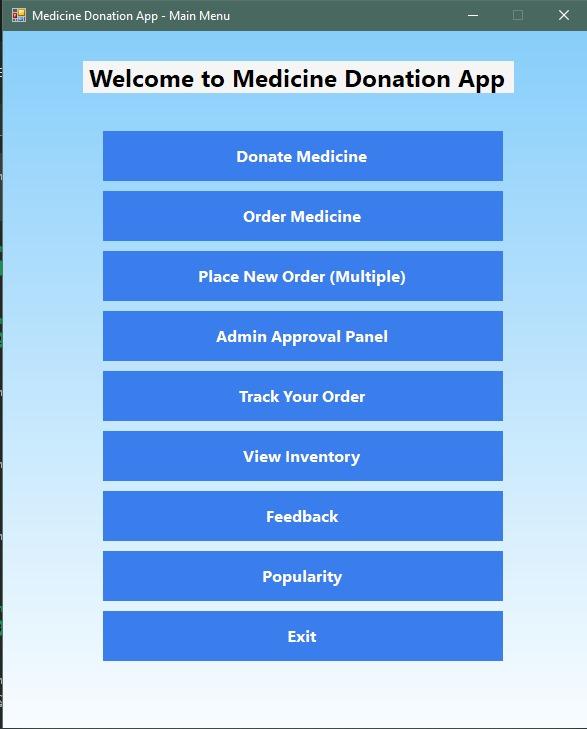
#### 3. **Data Layer – Model (Text File Storage)**

* Responsible for **data handling** and persistence.
* Uses plain **text files** to store:
  + Users
  + Medicines
  + Requests
  + Feedback
* Each data class (e.g., User.cs, Medicine.cs) represents a **Model** in the MVC pattern.

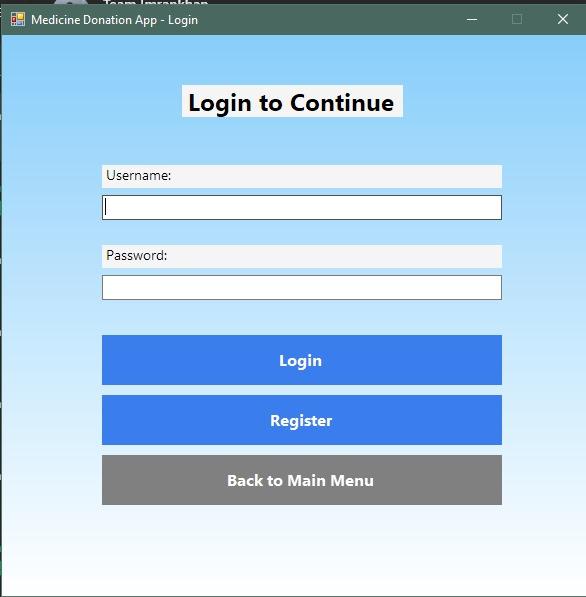


**e) Actual Implementation Screenshots**

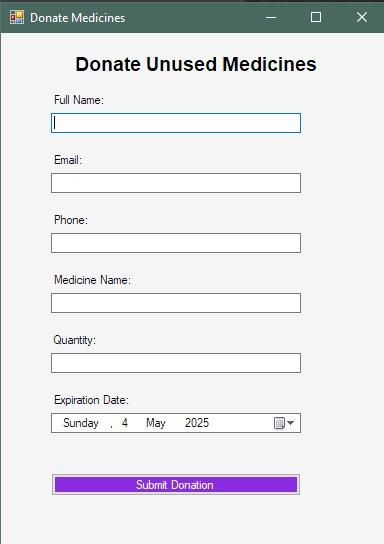
* Welcome Page



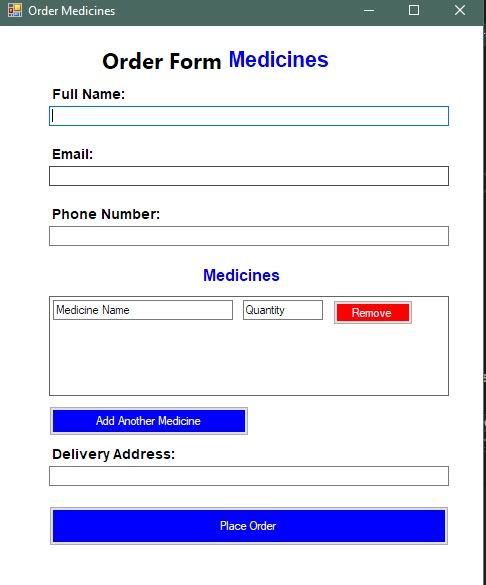
* Login and Registration Pages



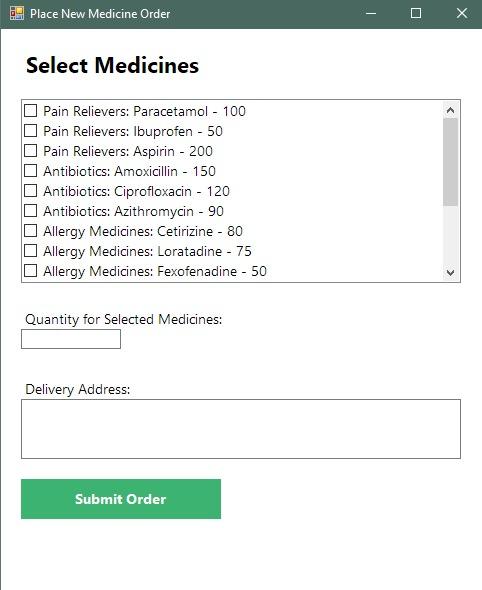
* Medicine Donation Form

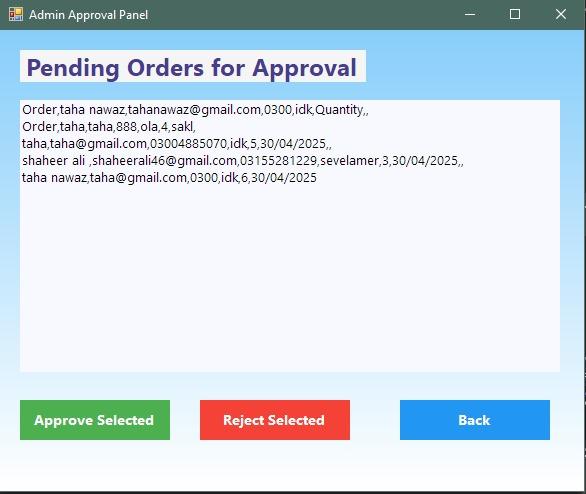


* Medicine Order Form

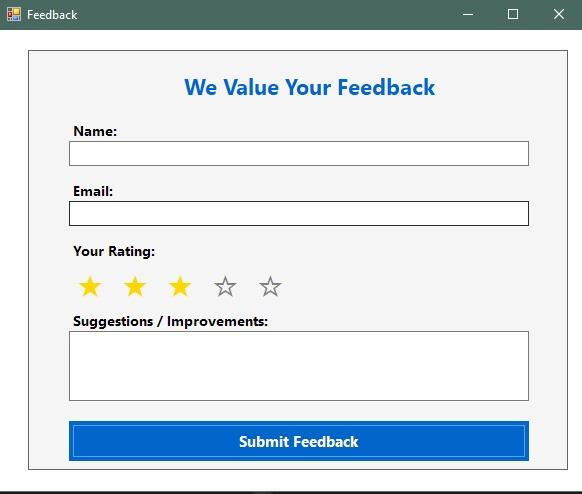


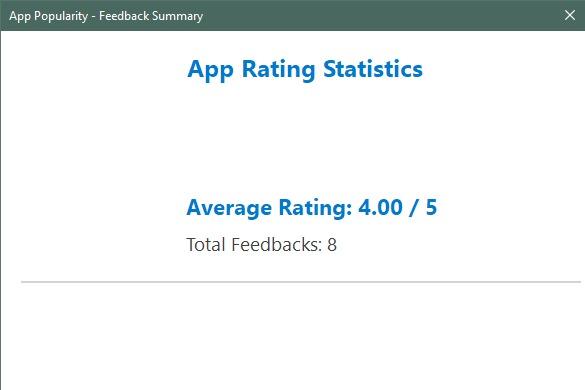
* Multiple Orders and Pending Orders Page



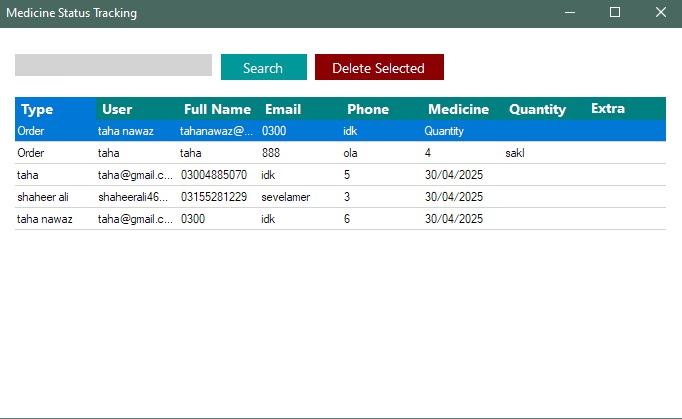


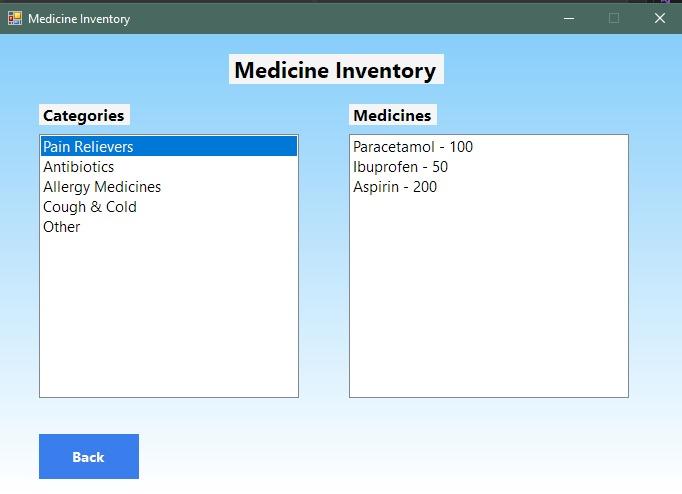
* Feedback Submission Page



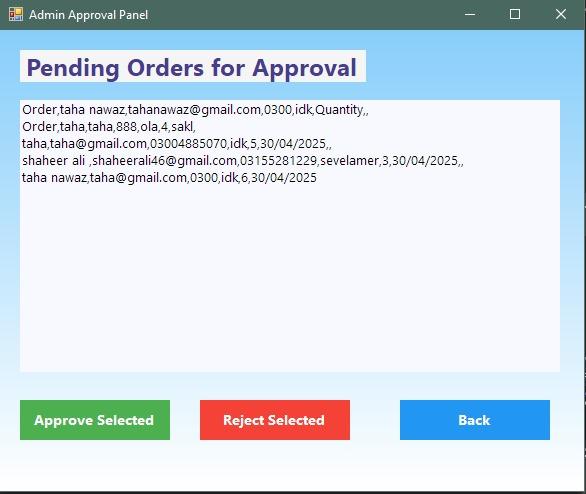


* Search History or Order History



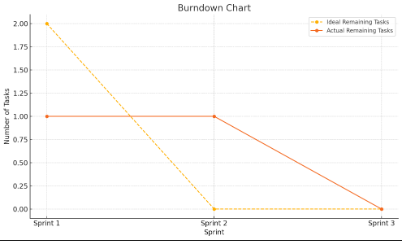


* Admin Approval Panel



**f) Product Burn Down Chart**

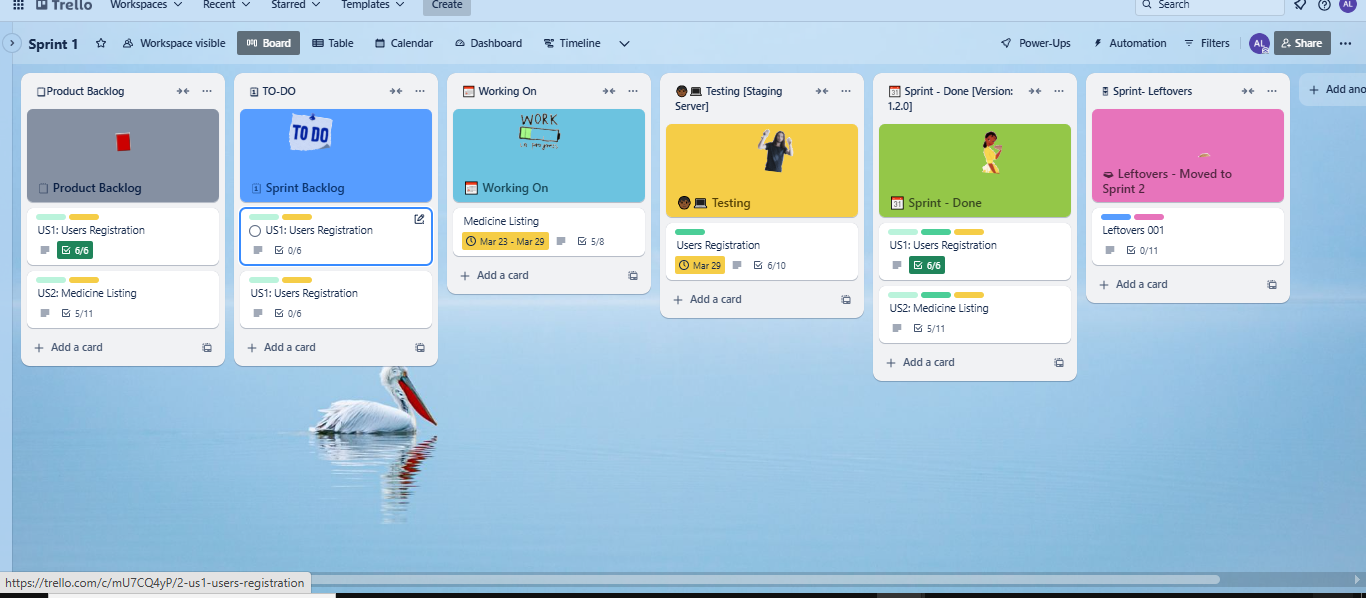
| **Sprint** | **Estimated Tasks** | **Completed Tasks** | **Remaining Tasks** |
| --- | --- | --- | --- |
| Sprint 1 | 2 | 1 | 1 |
| Sprint 2 | 4 (incl. leftover) | 3 | 1 |
| Sprint 3 | 6 (incl. leftovers) | 6 | 0 |



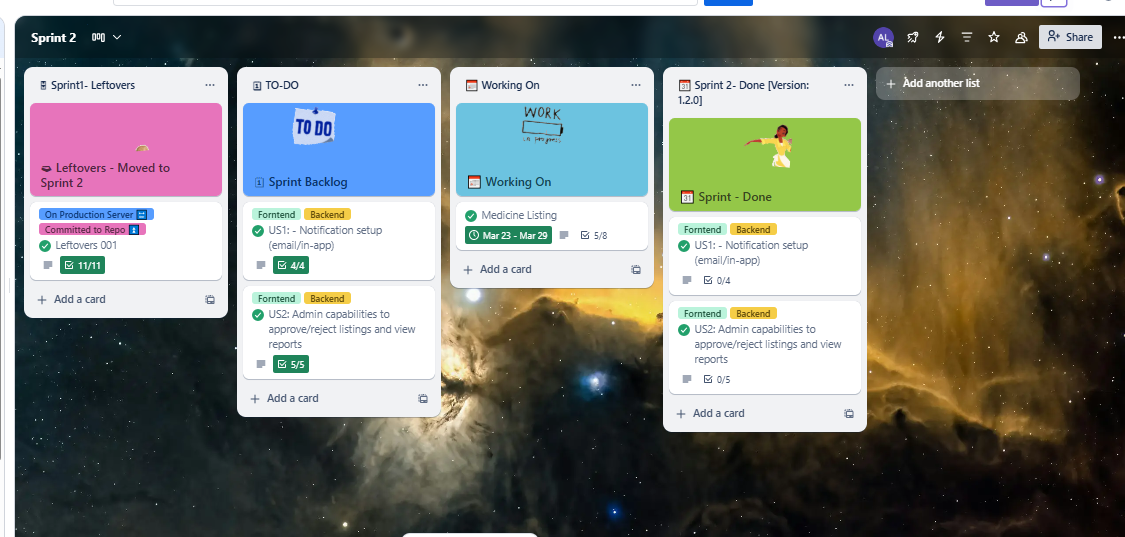
The dashed line shows the ideal progress, while the solid line shows actual task completion.

**g) Trello Board Screenshots**

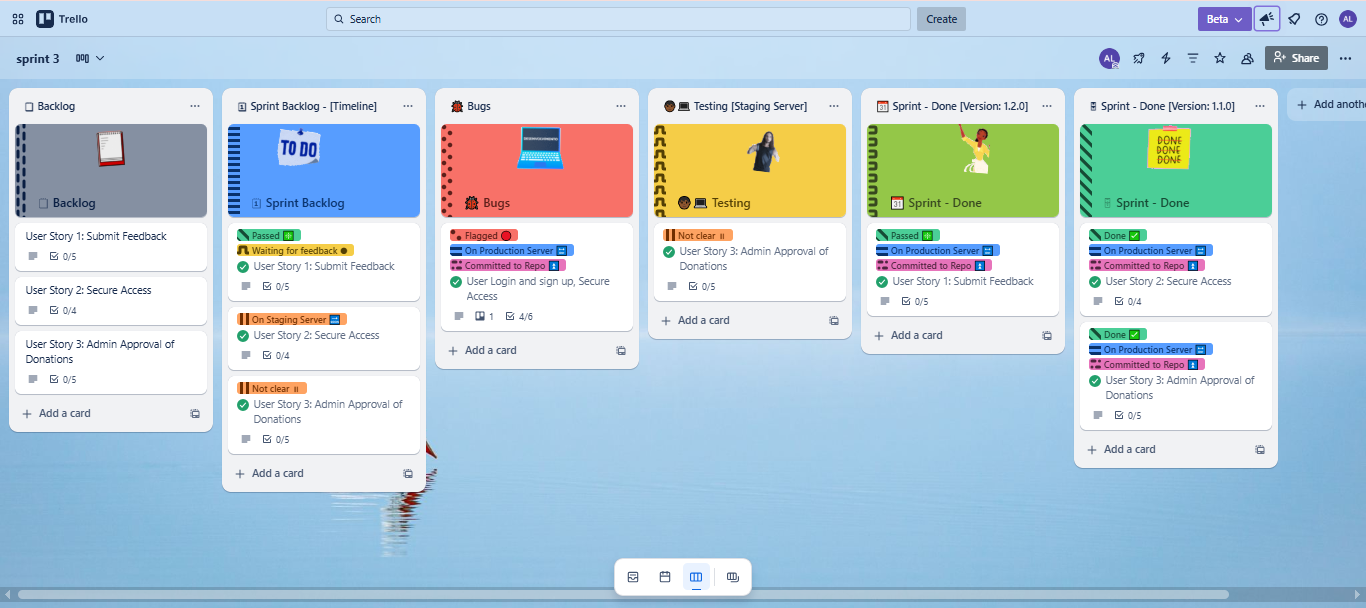
*Sprint 1*



*Sprint 2*



*Sprint 3*



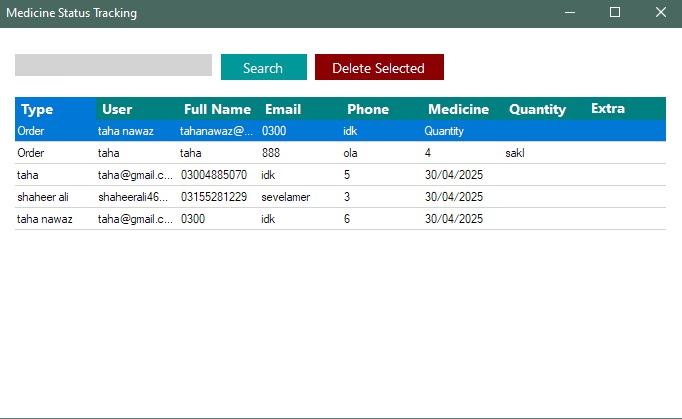
**h) Boundary Value Analysis Testing**

**Sign Up Testing**

| **Test Case ID** | **Field** | **Input** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| TC1 | Username | "" | Error: Required | Error shown | Pass |
| TC2 | Username | "usr" | Error: Too short | Error shown | Pass |
| TC3 | Username | "user" | Accepted | Accepted | Pass |
| TC4 | Username | "usernameusernameuserna" | Error: Too long | Error shown | Pass |
| TC5 | Password | "" | Error: Required | Error shown | Pass |
| TC6 | Password | "12345" | Error: Too short | Error shown | Pass |
| TC7 | Password | "abcdef" | Error: No number included | Error shown | Pass |
| TC8 | Password | "pass12" | Accepted | Accepted | Pass |
| TC9 | Password | 17-char input | Error: Too long | Error shown | Pass |
| TC10 | Email | "" | Error: Required | Error shown | Pass |
| TC11 | Email | "abc.com" | Error: Invalid format | Error shown | Pass |
| TC12 | Email | "[abc@yahoo.com](mailto:abc@yahoo.com)" | Error: Must end with @gmail.com | Error shown | Pass |
| TC13 | Email | "abc@gmail.com" | Accepted | Accepted | Pass |
| TC14 | Email | 51-char email | Error: Too long | Error shown | Pass |
| TC15 | Phone | "" | Error: Required | Error shown | Pass |
| TC16 | Phone | "123456789" | Error: Too short | Error shown | Pass |
| TC17 | Phone | "12345678901234" | Error: Too long | Error shown | Pass |
| TC18 | Phone | "12345abcde" | Error: Must be numeric | Error shown | Pass |
| TC19 | Phone | "1234567890" | Accepted | Accepted | Pass |
| TC20 | Age | 17 | Error: Underage | Error shown | Pass |
| TC21 | Age | 18 | Accepted | Accepted | Pass |

**Login Testing**

| **Test Case ID** | **Field** | **Input** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| TC22 | Username | "" | Error: Required | Error shown | Pass |
| TC23 | Password | "" | Error: Required | Error shown | Pass |
| TC24 | Credentials | Valid creds | Login Success | Success | Pass |



**i) Work Division Between Group Members**

| **Member Name** | **Responsibility** |
| --- | --- |
| Shahzeb | Frontend (Windows Forms Design) |
| Taha | Backend logic & database |
| Taha | Admin panel, feedback, login logic |
| Ali | Testing, documentation, Trello updates |

**j) Lessons Learned By Group**

* Applied Agile methodology (Trello and Sprints) to manage workflow effectively.
* Built a complete end-to-end secure web application using real-time login, validation.
* Learned the value of early architecture planning and requirement analysis.
* Strengthened teamwork, task ownership, and collaboration using GitHub and communication tools, and implemented both the frontend and backend of the system.