Introduction

Sprint 3 focused on implementing new functionality, including user authentication and authorization, hourly and daily forecast viewing, saving favorite locations, and push notifications for weather alerts. We followed agile development methodologies to ensure that the project is delivered on time and to the satisfaction of stakeholders. **This report provides an overview of the tasks completed during the sprint, as well as the results achieved.**

Sprint Backlog

The sprint backlog included four user stories, each with sub-user stories and tasks. The user stories were:

i. The remaining modules that will be developed in Sprint 3 are:

- User authentication and authorization functionality.
- Hourly and daily forecast viewing.
- Saving favorite locations.
- Push notifications for weather alerts.

ii. User Stories and respective sub-user stories to be developed in Sprint 3 (in proper user story template):

User Story 1: User Authentication and Authorization

As a user, I want to be able to create an account and log in so that I can access personalized weather data and settings.

- **Sub-User Story 1: Create User Accounts**
  As a developer, I want to create a database schema for user accounts (3 hours) and implement API endpoints for user registration (4 hours) and login (4 hours).

- **Sub-User Story 2: Authenticate and Authorize Users**
  As a developer, I want to implement user authentication and authorization functionality (8 hours) using industry-standard protocols and encryption methods.

User Story 2: Hourly and Daily Forecast Viewing

As a user, I want to be able to view both hourly and daily weather forecasts for any location.

- **Sub-User Story 1: Integrate Hourly and Daily Forecast Data Sources**
  As a developer, I want to integrate the application with one or more weather data sources that provide both hourly and daily forecasts (6 hours).

- **Sub-User Story 2: Display Hourly and Daily Forecast:**

As a developer, I want to implement UI components for displaying both hourly and daily weather forecasts (6 hours).

User Story 3: Saving Favorite Locations

As a user, I want to be able to save my favorite locations so that I can quickly access weather information for those locations.

- **Sub-User Story 1: Implement Location Saving**
  As a developer, I want to implement database schema for saved locations (3 hours) and implement API endpoints for adding (4 hours) and retrieving (4 hours) saved locations.

- **Sub-User Story 2: Display Saved Locations**
  As a developer, I want to implement UI components for displaying a list of saved locations (4 hours) and integrating saved locations with other parts of the application (4 hours).

User Story 4: Push Notifications for Weather Alerts

As a user, I want to receive push notifications for weather alerts that I have subscribed to.

- **Sub-User Story 1: Implement Push Notification Services:**
  As a developer, I want to implement a push notification service (8 hours) that can deliver alerts to users' devices.

- **Sub-User Story 2: Integrate Push Notifications with Alert System**
  As a developer, I want to integrate push notifications with the weather alert system (8 hours) so that users can receive alerts they have subscribed to via push notification.

iii. Respective Tasks of all the selected user stories and sub-user stories:

User Story 1: User Authentication and Authorization

- **Sub-User Story 1: Create User Accounts**

  - ✓ Implement database schema for user accounts (3 hours)
  - ✓ Implement API endpoint for user registration (4 hours)
  - ✓ Implement API endpoint for user login (4 hours)

- **Sub-User Story 2: Authenticate and Authorize Users**

  - ✓ Implement user authentication and authorization functionality (8 hours)

User Story 2: Hourly and Daily Forecast Viewing

- **Sub-User Story 1: Integrate Hourly and Daily Forecast Data Sources**
  - ✓ Research and select appropriate data sources (2 hours)
  - ✓ Implement data source integration (4 hours)

- **Sub-User Story 2: Display Hourly and Daily Forecasts**
  - ✓ Implement UI components for displaying hourly forecasts (4 hours)
  - ✓ Implement UI components for displaying daily forecasts (4 hours)

User Story 3: Saving Favorite Locations

- **Sub-User Story 1: Implement Location Saving**
  - ✓ Implement database schema for saved locations (3 hours)
  - ✓ Implement API endpoint for adding saved locations (4 hours)
  - ✓ Implement API endpoint for retrieving saved locations (4 hours)

- **Sub-User Story 2: Display Saved Locations**
  - ✓ Implement UI components for displaying a list of saved locations (4 hours)
  - ✓ Integrate saved locations with other parts of the application (4 hours)

User Story 4: Push Notifications for Weather Alerts

- **Sub-User Story 1: Implement Push Notification Service**

✓ Research and select appropriate push notification service (2 hours)
✓ Implement push notification service integration (6 hours)

- **Sub-User Story 2: Integrate Push Notifications with Alert System**

✓ Implement push notification subscriptions (4 hours)
✓ Implement push notification delivery for subscribed alerts (4 hours)

Results Achieved

During the sprint, the team was able to complete all the user stories and sub-user stories within the estimated time frame. The following were the results achieved for each user story:

- **User Authentication and Authorization**
  The team was able to implement user authentication and authorization functionality using industry-standard protocols and encryption methods. Users can now create an account, log in, and access personalized weather data and settings.

- **Hourly and Daily Forecast Viewing**
  The team integrated the application with one or more weather data sources that provide both hourly and daily forecasts. UI components for displaying both hourly and daily weather forecasts were also implemented.

- **Saving Favorite Locations**
  The team implemented a database schema for saved locations and API endpoints for adding and retrieving saved locations. UI components for displaying a list of saved locations were also implemented and integrated with other parts of the application.

- **Push Notifications for Weather Alerts**
  The team implemented a push notification service that can deliver alerts to users' devices. The push notification service was also integrated with the weather alert system so that users can receive alerts they have subscribed to via push notification.

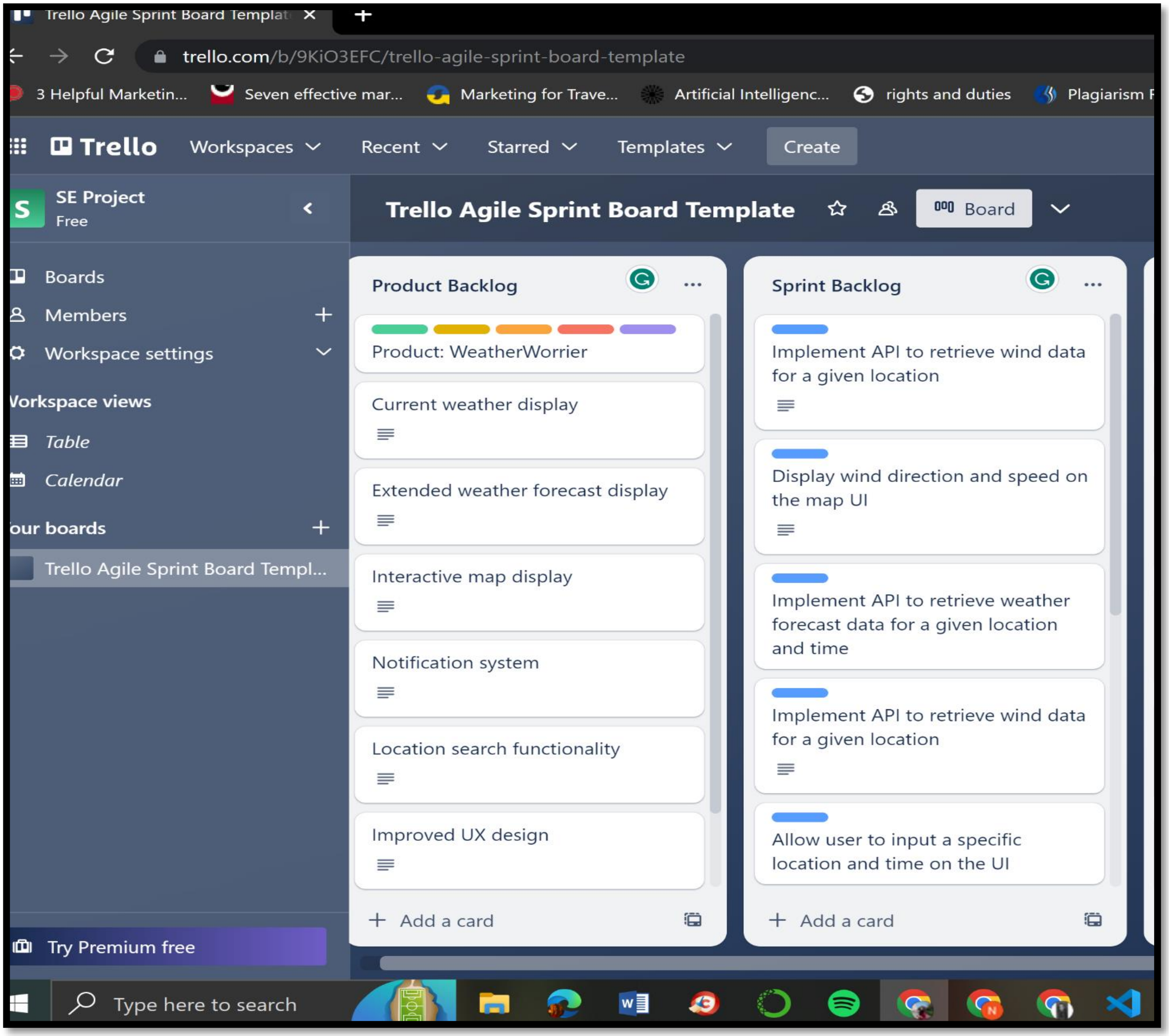
Test Cases

To ensure the functionality of the application, the team developed and executed test cases for each user story and sub-user story. The test cases covered all possible scenarios and inputs. Test cases were conducted manually and automatically using unit testing and integration testing techniques. The test results were documented and used to ensure the quality of the application.

Trello Board Snapshots

The Trello board was used to manage the project and track the progress of each task. The board was updated daily with the status of each task. The following are snapshots of the Trello board during the sprint:

1. **Snapshot-1** of prepared board (at the start of sprint) with product and sprint backlog on the board.



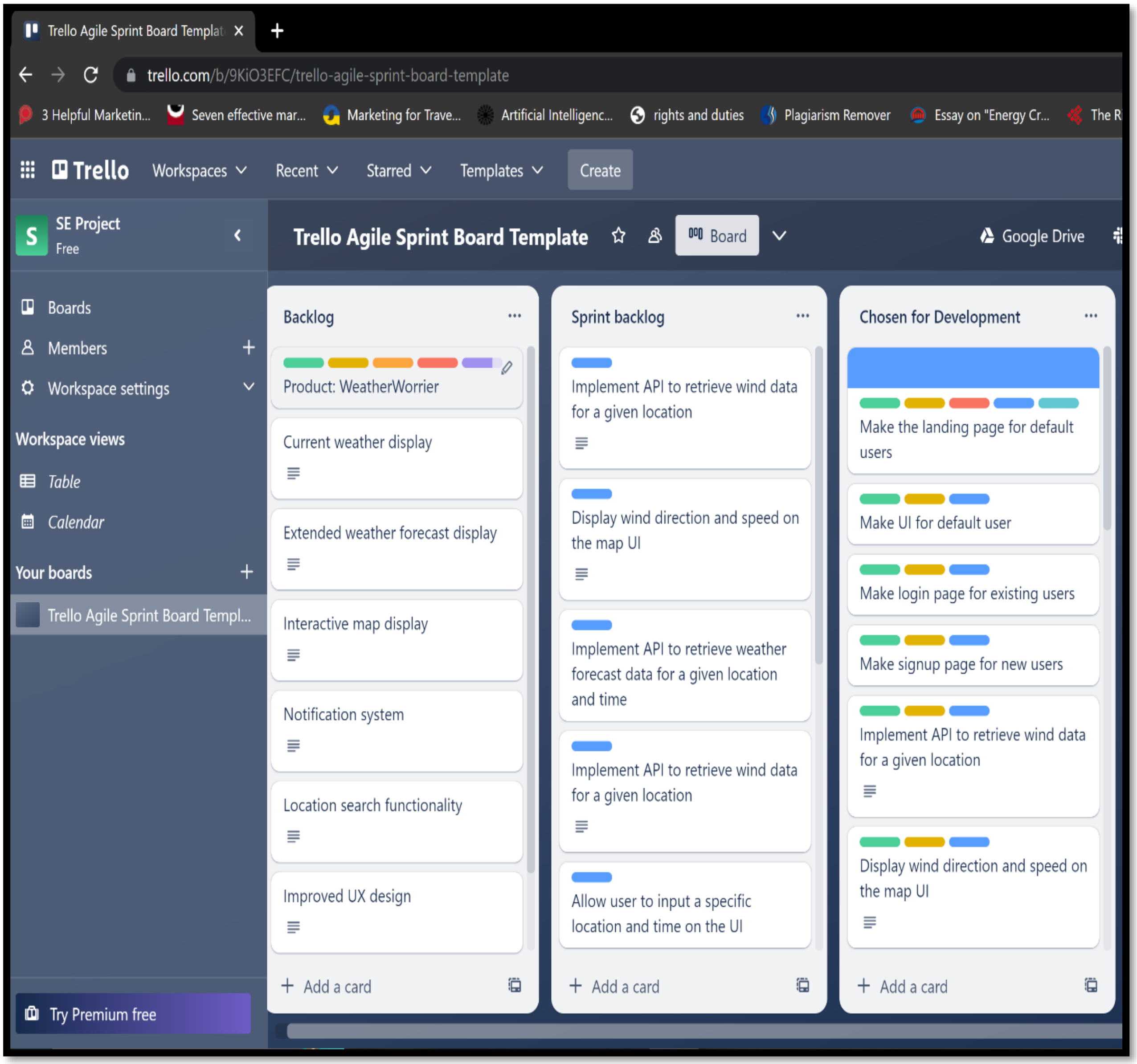
Product Backlog

- Product: WeatherWorrier
- Current weather display
- Extended weather forecast display
- Interactive map display
- Notification system
- Location search functionality
- Improved UX design

Sprint Backlog

- Implement API to retrieve wind data for a given location
- Display wind direction and speed on the map UI
- Implement API to retrieve weather forecast data for a given location and time
- Implement API to retrieve wind data for a given location
- Allow user to input a specific location and time on the UI

2. **snapshot-2** in the middle of the sprint, done with almost half of user stories and tasks.



Backlog

- Product: WeatherWorrier
- Current weather display
- Extended weather forecast display
- Interactive map display
- Notification system
- Location search functionality
- Improved UX design

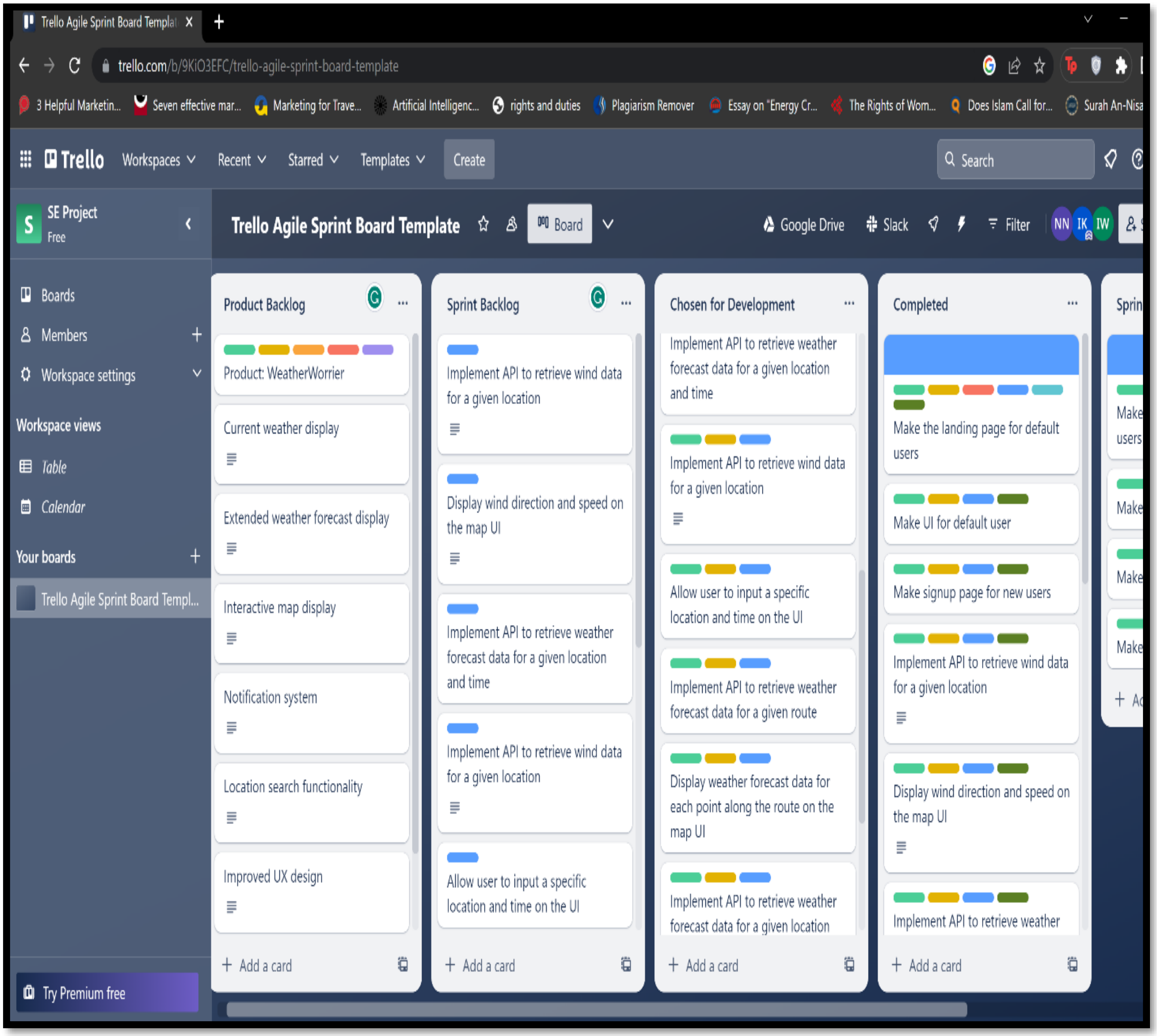
Sprint backlog

- Implement API to retrieve wind data for a given location
- Display wind direction and speed on the map UI
- Implement API to retrieve weather forecast data for a given location and time
- Implement API to retrieve wind data for a given location
- Allow user to input a specific location and time on the UI

Chosen for Development

- Make the landing page for default users
- Make UI for default user
- Make login page for existing users
- Make signup page for new users
- Implement API to retrieve wind data for a given location
- Display wind direction and speed on the map UI

3. **<u>snapshot-3</u>** at the end of the sprint when done with user stories and tasks.



Product Backlog

- Product: WeatherWorrier
- Current weather display
- Extended weather forecast display
- Interactive map display
- Notification system
- Location search functionality
- Improved UX design

+ Add a card

Sprint Backlog

- Implement API to retrieve wind data for a given location
- Display wind direction and speed on the map UI
- Implement API to retrieve weather forecast data for a given location and time
- Implement API to retrieve wind data for a given location
- Allow user to input a specific location and time on the UI

+ Add a card

Chosen for Development

- Implement API to retrieve weather forecast data for a given location and time
- Implement API to retrieve wind data for a given location
- Allow user to input a specific location and time on the UI
- Implement API to retrieve weather forecast data for a given route
- Display weather forecast data for each point along the route on the map UI
- Implement API to retrieve weather forecast data for a given location

+ Add a card

Completed

- Make the landing page for default users
- Make UI for default user
- Make signup page for new users
- Implement API to retrieve wind data for a given location
- Display wind direction and speed on the map UI
- Implement API to retrieve weather

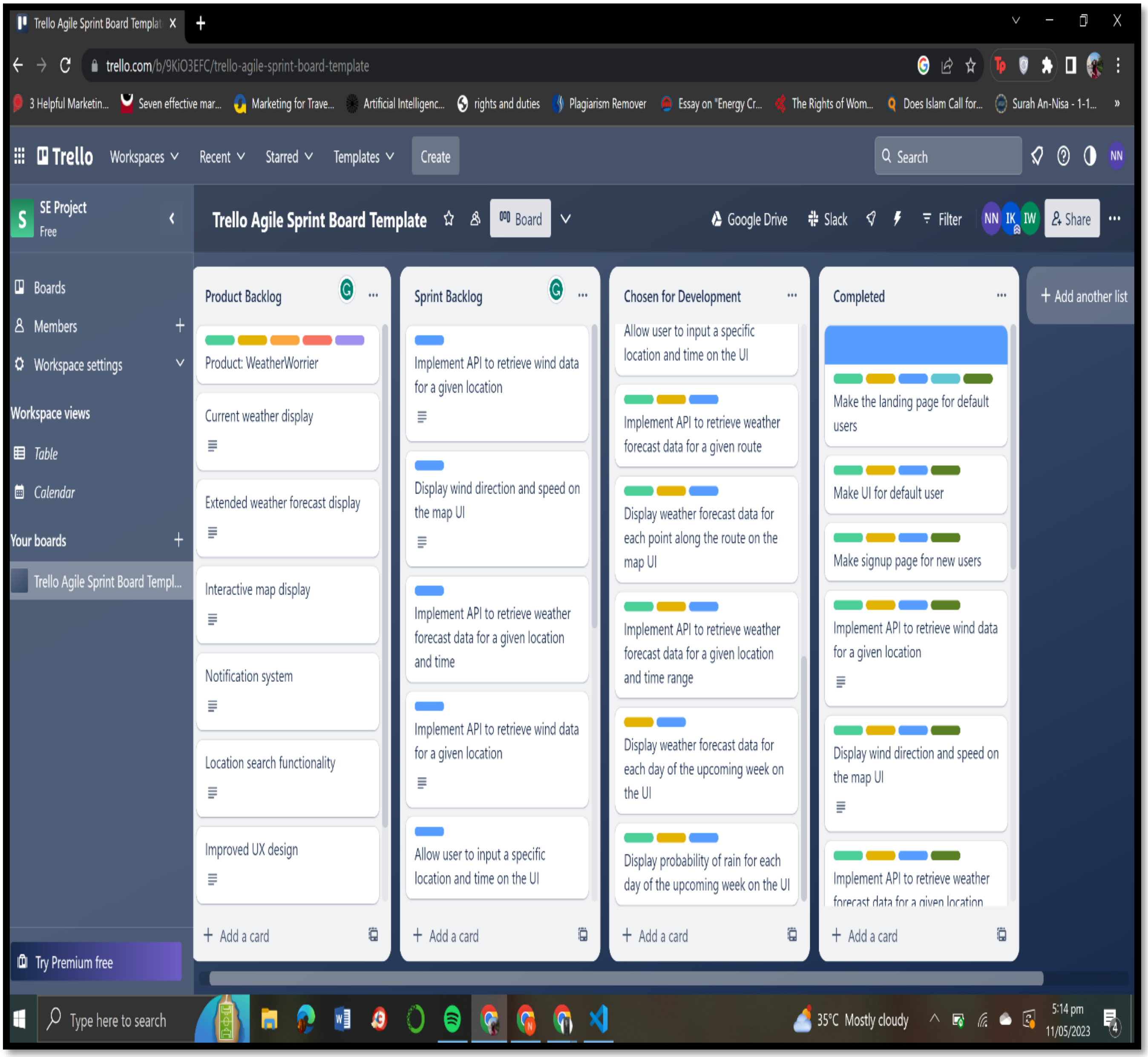
+ Add a card

Sprin

- Make users
- Make
- Make
- Make

+ Ad

4. **snapshot-4** complete trello board



Product Backlog

- Product: WeatherWorrier
- Current weather display
- Extended weather forecast display
- Interactive map display
- Notification system
- Location search functionality
- Improved UX design

Sprint Backlog

- Implement API to retrieve wind data for a given location
- Display wind direction and speed on the map UI
- Implement API to retrieve weather forecast data for a given location and time
- Implement API to retrieve wind data for a given location
- Allow user to input a specific location and time on the UI

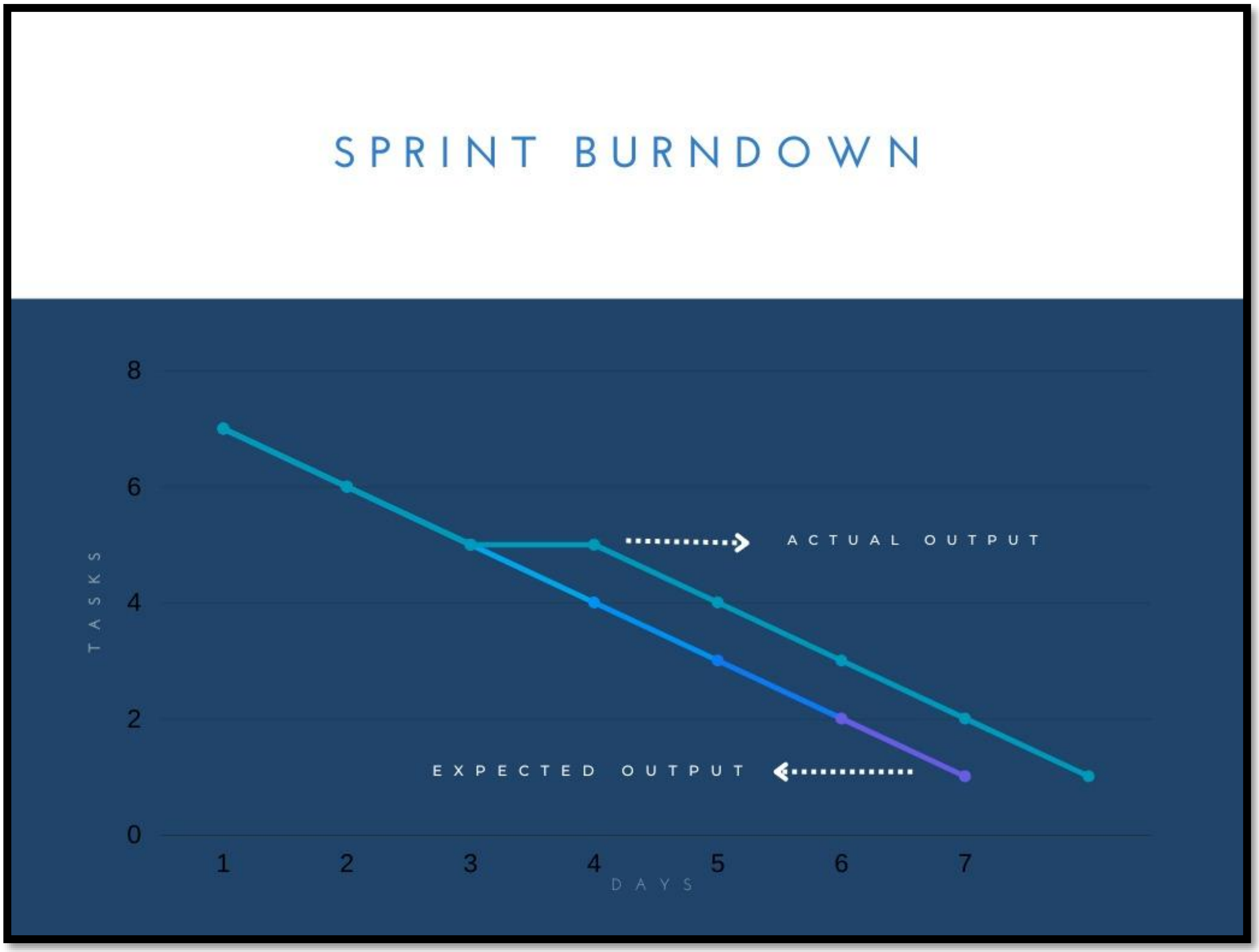
Chosen for Development

- Allow user to input a specific location and time on the UI
- Implement API to retrieve weather forecast data for a given route
- Display weather forecast data for each point along the route on the map UI
- Implement API to retrieve weather forecast data for a given location and time range
- Display weather forecast data for each day of the upcoming week on the UI
- Display probability of rain for each day of the upcoming week on the UI

Completed

- Make the landing page for default users
- Make UI for default user
- Make signup page for new users
- Implement API to retrieve wind data for a given location
- Display wind direction and speed on the map UI
- Implement API to retrieve weather forecast data for a given location

Sprint burnt down chart



Conclusion

Sprint 3 was successful in delivering new functionality to the application. The team was able to complete all the user stories and sub-user stories within the estimated time frame. Test cases were developed and executed to ensure the quality of the application. The Trello board was used to manage the project and track the progress of each task. The results achieved in Sprint 3 will be used as a foundation for future sprints to continue to develop the application.