Reflection Report: BackEND semester assignment

Section 1: Sprints, Epics, and Issues

Epics Overview

The project was organized into four epics to streamline the development process:

1. **Receptionist Dashboard:** Core functionality for staff check-in/out.
2. **Delivery Management System:** Manual entry and tracking of deliveries.
3. **User Interface Enhancements:** Application branding and user experience improvements.
4. **Project Documentation:** Sprint planning, reflection report, and README creation.

Why These Issues Were Chosen

The issues were derived to satisfy **WeDeliverTECH's functional requirements**:

- **Staff Check-in/Out:** Essential for attendance management.
- **Delivery Tracking:** Critical for managing and updating deliveries.
- **API Integration:** Provides dynamic and realistic staff data.
- **Toast Notifications:** Alert the receptionist for overdue tasks.
- **User Validation:** Ensures accurate and consistent input data.

Breakdown of Issues into Sprints:

Sprint **Issues**

Sprint 1 - Project setup (GitHub and folder structure)
- Create API integration for staff data
- Design dashboard layout (HTML/CSS)

Sprint 2 - Develop staff check-in/out functionality
- Validate input for return time and delivery driver entries
- Implement manual delivery board entry and table update

Sprint 3 - Implement toast notifications for overdue staff and deliveries
- Add vehicle icons for delivery driver visualization
- Add digital clock displaying live date/time

Sprint 4 - Testing and debugging of all features
- Write the Reflection Report (Jira.pdf)
- Prepare final README file and finalize submission

Section 2: Project Timeline, Board, and Backlog

Timeline

The project followed a **four-sprint structure** over four weeks:

- **Sprint 1:** Backend setup, API integration, and initial layout.
- **Sprint 2:** Core functionality development (staff tracking, delivery entry).
- **Sprint 3:** Feature enhancements (notifications, icons, clock).
- **Sprint 4:** Testing, debugging, and documentation preparation.

Progress Summary

- **Sprint 1:** Completed the foundation with successful API integration and UI layout.
- **Sprint 2:** Delivered staff and delivery tracking features with validated inputs.
- **Sprint 3:** Added interactive notifications, icons for deliveries, and a live digital clock.
- **Sprint 4:** Performed end-to-end testing, finalized documentation, and prepared for submission.

Screenshots

1. **Timeline View:** Displaying sprint progression and task completion.
2. **Active Sprint Board:** Tasks moved through **"To Do," "In Progress,"** and **"Done"** states.
3. **Backlog:** Clearly organized with priority-based tasks.

Section 3: Project Management Reflection

Planning and Organization

The Scrum methodology using Jira enabled:

- **Clear Task Prioritization:** Epics were divided into smaller, manageable issues.
- **Progress Visibility:** Active sprints showed task movement, reducing bottlenecks.
- **Incremental Delivery:** Features were developed iteratively, ensuring continuous feedback.

Feature Highlights

Staff Management Table:

  - **Design:** Displays staff name, picture, and status.
  - **Functionality:** Allows check-in/out with calculated return times.

Delivery Board:

  - **Input Validation:** Ensures all required fields (e.g., vehicle type, return time) are filled accurately.
  - **Icons:** Visual differentiation of cars and motorcycles for clarity.

Toast Notifications:

  - **Behavior:** Alerts appear once when staff or drivers exceed their expected return time.
  - **Purpose:** Ensures the receptionist stays informed about late returns.

Digital Clock:

- **Dynamic Display:** Updates every second using the setInterval() method.
- **Format:** Follows "Day, Month, Year, Hour:Minute:Second" for clarity.

Challenges and Solutions

Challenge	Solution
API Integration and Object Conversion	Used **JavaScript classes** to convert and store API responses dynamically into usable objects.
Toast Notification Timing	Implemented event listeners to display notifications only once and allowed user dismissals.
Input Validation for Return Time	Developed regex-based validation for hh:mm format with error prompts.
Dynamic Clock Performance	Optimized setInterval usage to ensure smooth, real-time clock updates.

Technologies Used

- **Frontend:** HTML5, CSS3, Bootstrap for layout and styling.
- **Scripting:** JavaScript (ES6+), jQuery for DOM manipulation.
- **APIs:** RandomUser API for generating dynamic staff data.

Achievements

1. Developed a fully functional **Reception Management Dashboard** meeting WeDeliverTECH's requirements.
2. Integrated real-time **toast notifications** for overdue tasks.
3. Added a **dynamic digital clock** and ensured UI consistency with company branding.
4. Delivered clear project management documentation, including sprints, issues, and progress tracking.

Learning Outcomes

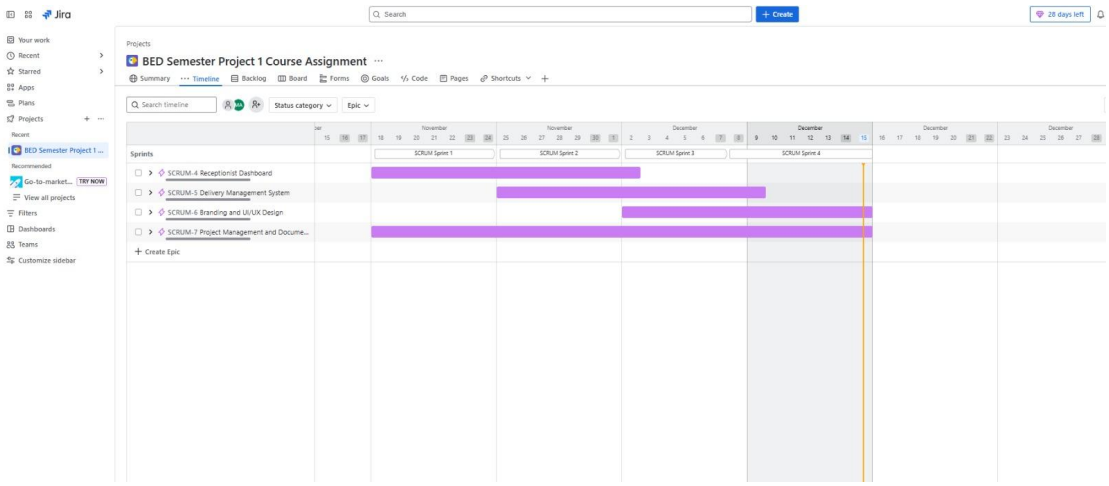
- Enhanced proficiency in **Jira Scrum methodology** for managing sprints and tasks.
- Improved technical skills in **JavaScript OOP** and **dynamic UI updates**.
- Gained experience in integrating APIs and handling **input validations** effectively.
- Learned to tackle challenges iteratively through sprint-based progress.

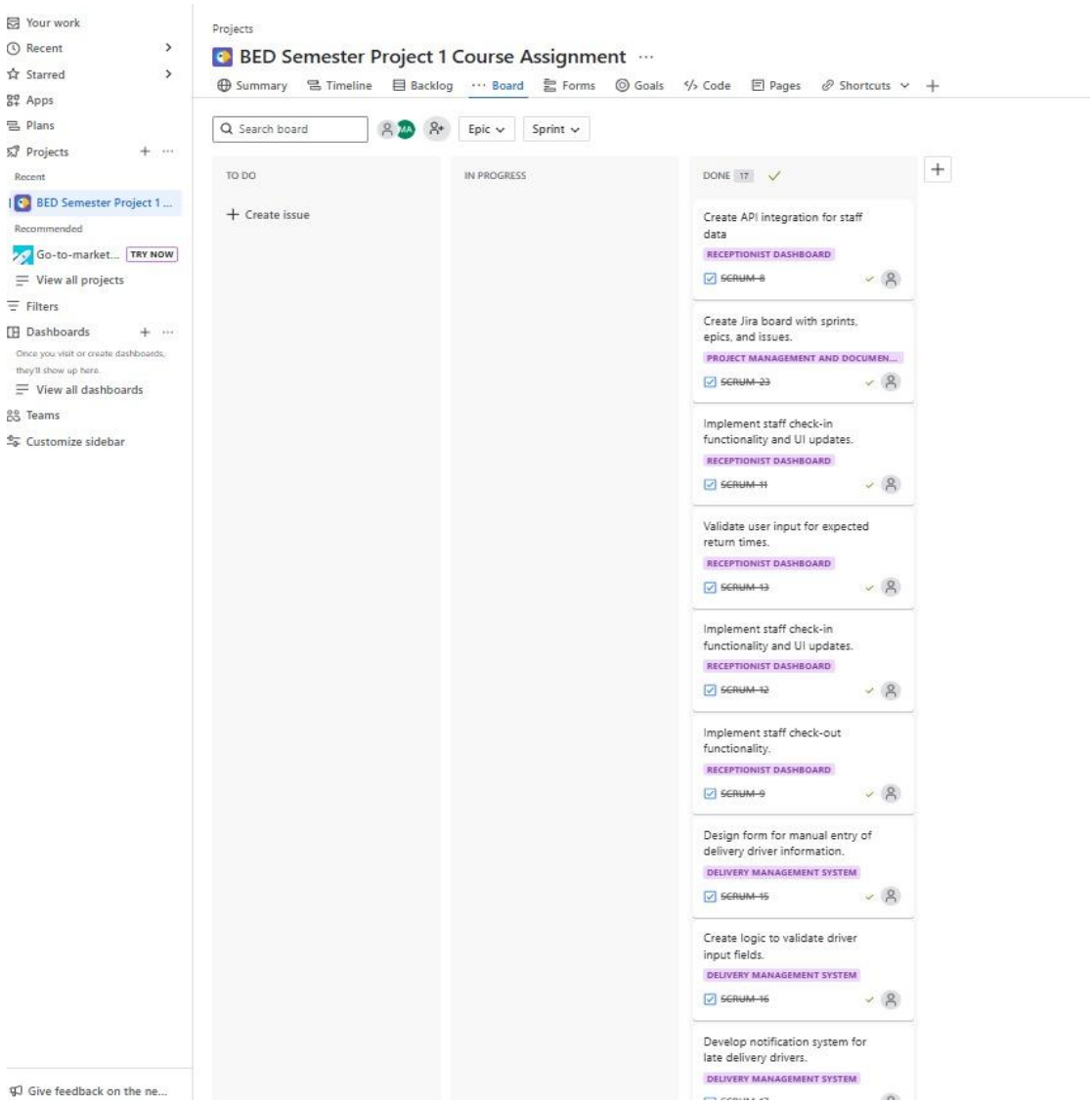
References

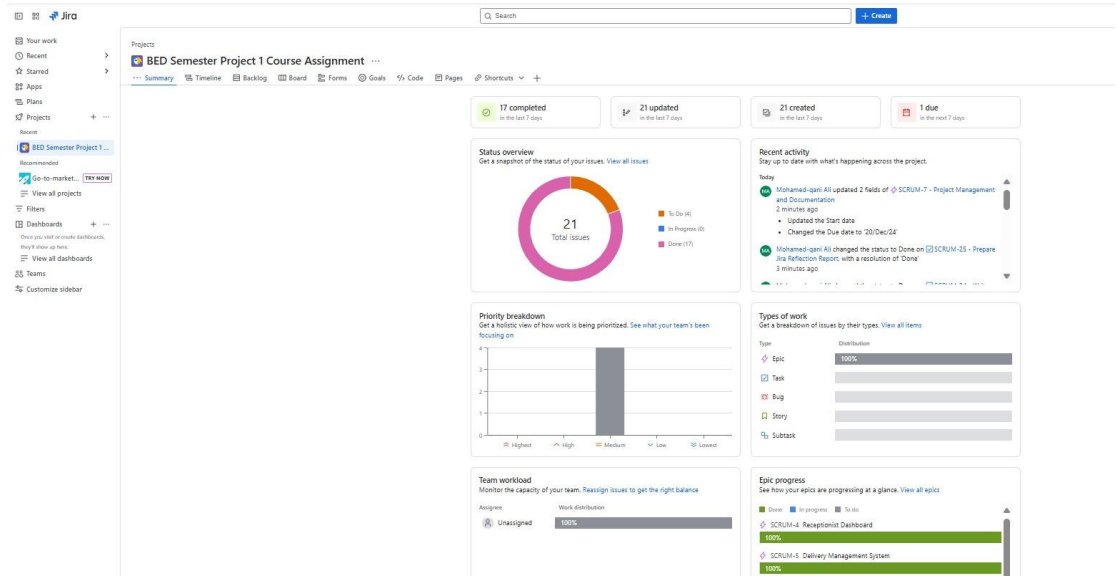
1. Beck, K., et al. (2001). *Manifesto for Agile Software Development*. Available at: Agile Manifesto.
2. MDN Web Docs (2024). *JavaScript Guide*. Available at: MDN Docs.
3. Bootstrap Framework (2024). *Introduction to Bootstrap*. Available at: Bootstrap Docs.

Appendix: Screenshots

- Jira timeline view with sprint breakdowns.
- Active sprint board showing task movement.
- Backlog overview with prioritized issues.




Your work
Recent
Starred
Apps
Plans
Projects
Recent
BED Semester Project 1 ...
Recommended
Go-to-market...
TRY NOW
View all projects
Filters
Dashboards
Once you visit or create dashboards, they'll show up here.
View all dashboards
Teams
Customize sidebar
Give feedback on the ne...
Projects
BED Semester Project 1 Course Assignment
Summary
Timeline
Backlog
Board
Forms
Goals
Code
Pages
Shortcuts
Search board
Epic
Sprint
TO DO
Create issue
IN PROGRESS
DONE 17
Create API integration for staff data
RECEPTIONIST DASHBOARD
SCRUM-8
Create Jira board with sprints, epics, and issues.
PROJECT MANAGEMENT AND DOCUMEN...
SCRUM-23
Implement staff check-in functionality and UI updates.
RECEPTIONIST DASHBOARD
SCRUM-11
Validate user input for expected return times.
RECEPTIONIST DASHBOARD
SCRUM-13
Implement staff check-in functionality and UI updates.
RECEPTIONIST DASHBOARD
SCRUM-12
Implement staff check-out functionality.
RECEPTIONIST DASHBOARD
SCRUM-9
Design form for manual entry of delivery driver information.
DELIVERY MANAGEMENT SYSTEM
SCRUM-15
Create logic to validate driver input fields.
DELIVERY MANAGEMENT SYSTEM
SCRUM-16
Develop notification system for late delivery drivers.
DELIVERY MANAGEMENT SYSTEM


Jira
Search
Create
Projects
BED Semester Project 1 Course Assignment
Summary
Timeline
Backlog
Board
Forms
Goals
Code
Pages
Shortcuts
17 completed
21 updated
21 created
1 due
Status overview
Get a snapshot of the status of your issues. View all issues
21
Total issues
To Do (4)
In Progress (0)
Done (17)
Recent activity
Stay up to date with what's happening across the project.
Today
Mohamed-qani Ali updated 2 fields of SCRUM-7 - Project Management and Documentation
2 minutes ago
Updated the Start date
Changed the Due date to '20/Dec/24'
Mohamed-qani Ali changed the status to Done on SCRUM-25 - Prepare Jira Reflection Report, with a resolution of 'Done'
3 minutes ago
Priority breakdown
Get a holistic view of how work is being prioritized. See what your team's been focusing on
Highest
High
Medium
Low
Lowest
Types of work
Get a breakdown of issues by their types. View all items
Type
Distribution
Epic
100%
Task
Bug
Story
Subtask
Team workload
Monitor the capacity of your team. Reassign issues to get the right balance
Assignee
Work distribution
Unassigned
100%
Epic progress
See how your epics are progressing at a glance. View all epics
Done
In progress
To do
SCRUM-4 Receptionist Dashboard
100%
SCRUM-5 Delivery Management System
100%