

Daily Scrum (1)

[20/11/2025] -Sprint 1

Members:

Omar Mohamed Mostafa 22p0197

Ahmed Wael Rafaat - 22p0221

Roaa Sherif gadara – 22p0188

Youssef Amr Farouk – 2200844

Ezzeldin Ismail kaoud – 22p0141

Sprint Goals: (total points 23)

US 1.1 - View Room Availability **DONE**

- Priority: Must Have
- Story Points: 5
- As a student/professor, I want to view available classrooms/labs so that I can find a free room.
- **T1.1.1 – DB design for rooms and availability (Assigned to Omar) **DONE****
 - Basic fields: roomid , name, type, capacity, status
- **T1.1.3 – Rooms page UI (Assigned to Roaa) **DONE****
 - “Rooms” page with table showing all rooms
- **T1.1.4 – Connect UI – backend (Assigned to Youssef) **DONE****
 - develop controllers for ui and connect database

US 1.3 - Manage Classroom / Lab records

- Priority: Must Have
- Story Points: 5
- As a professor/staff, I want to book a room so I can schedule classes/events.
- **T1.3.1 – Admin-only-access check (Assigned to Youssef) **70%****
 - make sure only admin can access this page
- **T1.3.2 – Backend CRUD for rooms (Assigned to Youssef) **DONE****
 - Create backend endpoints and services for:

- Create classroom/lab
- Update classroom/lab
- Delete classroom/lab
- **T1.3.3 – Admin UI (Assigned to Roaa) *DONE***
 - - Create admin interface with:
 - Page/form to add rooms
 - Page/form to edit rooms
 - Page/form to delete rooms
- **T1.3.4 – Testing (Assigned to Youssef/Roaa) *70%***
 - Verify:
 - Admin can perform CRUD operations properly
 - Non-admins cannot access the functionality

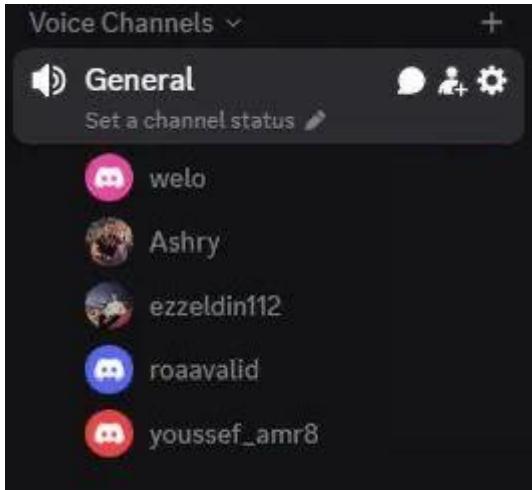
US 1.6 – Report Maintenance Issue

- Priority: Must Have
- Story Points: 5
- As a student/professor/staff, I want to track ticket progress.
- **T1.6.1 – DB design for maintenance tickets (Assigned to Omar) *DONE***
 - Basic fields: Ticket ID , Reporter, Room, Description, Status , Created date
- **T1.6.2 – Backend to create maintenance ticket (Assigned to Ezzeldin) *DONE***
 - Create backend endpoint/service for submitting maintenance tickets
- **T1.6.3 – UI form to submit maintenance issue (Assigned to Ahmed) *DONE***
 - User interface for submitting reports:
 - "Report Issue" page
- **T1.6.4 – Testing (Assigned to Ezzeldin/Omar) *70%***
 - Verify:
 - Logged-in users (student/staff/professor) can submit tickets
 - Users see success message after submission

US 4.1 – User Authentication (Login)

- Priority: Must Have
- Story Points: 5
- As a user, I want to log in based on my role, so that I can use services available to me based on my role.
- **T4.1.1 – Design DB structure for users & roles (Assigned to Omar) *DONE***
 - Basic fields: User table/entity , Role table/entity, Room, UserRole mapping table
- **T4.1.2 – Implement backend login logic (Assigned to Ezzeldin) *80%***
 - Backend authentication service:
 - Login endpoint with username/password
 - Role verification
 - Permission loading based on role
- **T4.1.3 – Implement login UI (Assigned to Ahmed) *DONE***
 - User interface for submitting reports:
 - Login form (username/email + password fields)
 - Validation messages for different error scenarios
- **T4.1.4 – Role-based access check (basic) (Assigned to Ezzeldin/Omar) *70%***
 - - Post-login routing:
 - - Redirect users to appropriate dashboards based on role: (Student , Admin , Staff ,Professor)
- **T4.1.6 – Test login (Assigned to Ahmed) *70%***
 - Comprehensive testing
 - Unit tests
 - Manual test scenarios
 - Coverage based on acceptance criteria

Screenshots



Plan for next 24 hours: (all to finish 100%)

- **Omar Mohamed Mostafa**
 - T4.1.4 – Role-based access check (basic)
 - T1.6.4 – Testing

- **Ahmed Wael Rafaat**
 - T4.1.6 – Test login
- **Roaa Sherif Gadara**
 - T1.3.4 – Testing

- **Youssef Amr Farouk**
 - T1.3.4 – Testing (Assigned to Youssef)
 - T1.3.1 – Admin-only-access check (Assigned to Youssef)

- **Ezzeldin Ismail Kaoud**
 - T4.1.4 – Role-based access check (basic)
 - T4.1.2 – Implement backend login logic (rest of the implementation)
 - T1.6.4 – Testing