

PayTrack

Rio.co

Noor Fatima – 22i-1036

Sara Akbar – 22i-0846

Abdul Munim – 22i-1080

| | |
|---|-----------|
| Project Brief..... | 1 |
| Functional Requirements and non-Functional Requirements..... | 2 |
| Functional Requirements..... | 2 |
| Non-functional Requirements..... | 2 |
| Project Vision..... | 3 |
| User Stories and sub user stories..... | 3 |
| Employee Leave Request..... | 3 |
| Product Backlog..... | 8 |
| Sprint Backlog..... | 8 |
| Project Plan..... | 10 |
| Project Initiation..... | 10 |
| Architecture & Setup..... | 11 |
| Backend Development (Spring Boot)..... | 11 |
| Frontend Development (ReactJS)..... | 11 |
| Testing..... | 12 |
| Final Deliverables..... | 12 |
| Architecture Diagram..... | 12 |
| Design (sprint 3)..... | 13 |
| Actual implementation Screenshot..... | 13 |
| Product burndown chart (complete project)..... | 16 |
| Trello Board Screenshots..... | 16 |
| Test Cases Blackbox..... | 20 |
| Test Cases whitebox..... | 21 |
| Backend..... | 21 |
| What is Covered Well..... | 21 |
| What is Not Covered and Why..... | 22 |
| Frontend..... | 22 |
| Software Engineering Project Work Division..... | 22 |
| Sprint 1: Frontend Development..... | 22 |
| Module Development..... | 23 |
| Backend Development..... | 23 |
| User Stories and Task Management..... | 23 |
| Documentation..... | 23 |
| Testing..... | 23 |
| Lessons Learned from Software Engineering Project..... | 23 |

Project Brief

PayTrack is a Leave and Payroll Management System designed to automate salary processing, leave tracking, and employee management. It ensures accurate payroll calculations based on deductions and leave data while allowing employees to apply for leave, track approvals, and access payslips. Managers can approve or reject leave requests and monitor leave balances to ensure policy compliance. HR personnel handle employee registration, payroll processing, and leave reports, offering insights into leave trends and salary adjustments. Administrators oversee HR and managers, ensuring structured access control and efficient management. By reducing manual workload, improving accuracy, and streamlining payroll and leave approvals, PayTrack enhances organizational efficiency and data-driven decision-making.

Functional Requirements and non-Functional Requirements

Functional Requirements

- The system shall allow employees to submit leave requests specifying leave type, start date, end date, and reason.
- The system shall allow managers to approve or reject leave requests.
- The system shall allow managers to view employees' leave balances before approving requests.
- The system shall allow employees to check the approval status of pending leave requests.
- The system shall calculate employee salaries based on leave records, and applicable deductions.
- The system shall process payroll on a monthly basis.
- The system shall allow HR to remove employees from the system.
- The system shall allow administrators to add/remove HR personnel and managers.

Non-functional Requirements

- ***System Non-Functional Requirements***
- The system shall generate and distribute payslips within 2 seconds after employee checks for their pay.
- The system shall respond to employee leave requests within 1 second after submission.
- The system shall support the addition of employees and departments without degrading performance.
- The system shall ensure that leave requests are updated in real-time on employee and manager dashboards.

➤ **Organizational requirements**

- The system shall be maintainable and allow feature updates without affecting core functionalities.
- The system shall allow HR to configure salary structures, leave policies, and deductions as per company rules.
- The system shall generate leave reports for HR and management to analyze workforce trends.

➤ **External Requirements**

- The system shall require secure authentication methods.

Project Vision

- The vision of PayTrack is to develop a robust, user-friendly, and efficient Leave and Payroll Management System that streamlines salary processing, leave tracking, and employee management while ensuring accuracy and reducing administrative workload.
- PayTrack aims to:
- Automate payroll calculations based on deductions and leave data for error-free salary processing.
- Enable seamless pay slip generation and timely salary distribution.
- Allow employees to apply for leave, track approvals, and view their leave balance in real time.
- Empower managers to approve/reject leave requests and track leave balances to ensure compliance.
- Provide HR with tools to add employees, generate payroll, and create leave reports for insights on leave trends.
- Give administrators control over HR and manager roles, ensuring structured access and management.
- By integrating these features, PayTrack enhances efficiency, improves decision-making, and ensures a seamless payroll and leave management experience for organizations.

User Stories and sub user stories

Employee Leave Request

■ **User Story:**

- *"As an employee, I want to request leave so that I can take time off when needed."*

■ **Sub-User Stories:**

- As an employee, I want to select the type of leave (casual, sick, paid, unpaid) so that my request is categorized correctly.

- As an employee, I want to provide a reason for my leave request so that my manager understands why I need time off.
- As an employee, I want to see my remaining leave balance before submitting my request.

Leave Approval by Manager

■ User Story:

- *"As a manager, I want to approve or reject leave requests so that I can manage team availability."*

■ Sub-User Stories:

- As a manager, I want to see all pending leave requests on my dashboard.
- As a manager, I want to view reports on leave trends to analyze team leave patterns.

Payroll processing

■ User Story:

- *"As an HR manager, I want to process employee payroll so that salaries are disbursed on time."*

■ Sub-User Stories:

- As an HR manager, I want to calculate salaries based on leave records.
- As an HR manager, I want to apply deductions (e.g., taxes, benefits, unpaid leave) automatically.
- As an HR manager, I want to generate monthly salary slips for all employees.

Employee Salary access slip

■ User Story:

- *"As an employee, I want to view my salary slip so that I can keep track of my earnings."*

■ Sub-User Stories:

- As an employee, I want to access my salary slip for any past month.
- As an employee, I want to see a breakdown of my salary (basic pay, allowances, deductions).
- As an employee, I want to request corrections if there is an error in my salary slip.

View Employee profile

■ User Story:

- *"As an employee, I want to view and update my profile so that my information remains up to date."*

■ Sub-User Stories:

- As an employee, I want to change my password to maintain account security.
- As an employee, I want to update my contact information (phone, address) so that HR has my latest details.
- As an employee, I want to view my leave history so that I can track my past leave records.

Leave Balance Tracking

■ **User Story:**

- *"As a manager, I want to track employees' leave balances so that I can decide if they are eligible for more leave."*

■ **Sub-User Stories:**

- As a manager, I want to see an employee's remaining leave days before approving a request.
- As a manager, I want to generate a leave balance report for my team to monitor their available leave.

HR Employee Onboarding

■ **User Story:**

- *"As an HR manager, I want to add new employees to the system so that their details are recorded for payroll and leave management."*

■ **Sub-User Stories:**

- As an HR manager, I want to assign employees to departments upon adding them.
- As an HR manager, I want to set an employee's initial leave balance upon hiring.
- As an HR manager, I want to update an employee's role or department if they get promoted or transferred.

HR Employee Removal

■ **User Story:**

- *"As an HR manager, I want to remove employees from the system so that records remain up to date and payroll and leave data are properly managed."*

■ **Sub-User Stories:**

- As an HR manager, I want to deactivate an employee's account instead of immediate deletion for record-keeping purposes.
- As an HR manager, I want to receive a confirmation prompt before finalizing an employee's removal.

Employee Leave Approval Status

■ **User Story:**

- *"As an employee, I want to check the approval status of my pending leave requests so that I know whether my leave has been approved or rejected."*

■ **Sub-User Stories:**

- As an employee, I want to view all my pending leave requests in a dedicated section so that I can track their status easily.
- As an employee, I want to see the reason for rejection if my leave is not approved so that I understand the decision.
- As an employee, I want to filter my pending leave requests by date or leave type so that I can easily locate specific requests.

HR Leave Reporting

■ **User Story:**

- *"As an HR manager, I want to generate leave reports that summarize each employee's leave usage and averages so that I can effectively monitor leave patterns and manage workforce planning."*

■ **Sub-User Stories:**

- As an HR manager, I want the system to calculate the total number of leaves taken by each employee over a specified period so that I can identify leave trends.
- As an HR manager, I want the system to compute the average number of leaves per employee to assess overall team leave behavior.
- As an HR manager, I want to generate a comprehensive report that includes details such as leave types, totals, and averages, so that I can easily review and analyze the data.

Add HR and Manager

■ **User Story:**

- *"As an admin, I want to add new HR managers and department managers so that they can manage employees effectively."*

■ **Sub-User Stories:**

- As an admin, I want to assign roles (HR or Manager) when adding a new user to the system.
- As an admin, I want to set login credentials for newly added HR managers and managers.

- As an admin, I want to receive a confirmation message when a new HR or manager is successfully added.

Remove HR and Manager

■ **User Story:**

- *"As an admin, I want to remove HR managers and department managers so that the system stays up-to date with active staff."*

■ **Sub-User Stories:**

- As an admin, I want to deactivate an HR or manager's account when they leave the company.
- As an admin, I want to prevent accidental removals by adding a confirmation step before deletion.

View List of HR and Manager

■ **User Story:**

- *"As an admin, I want to view a list of all HR managers and department managers so that I can track assigned personnel."*

■ **Sub-User Stories:**

- As an admin, I want to filter the list based on role (HR or Manager).
- As an admin, I want to see details such as name, assigned employees, and date of joining.

Employee Login and Authentication

● **User Story:**

- *"As an employee, I want to log in to the system securely so that I can access my personalized dashboard and perform my job-specific tasks."*

● **Sub-User Stories:**

- As an employee, I want to be able to change my username and password through a secure settings page, so that I can maintain control of my login credentials.
- As an employee, I want the login process to be fast and responsive, ensuring that I can access the system quickly and without delays

Manager Login and Authentication

■ **User Story:**

- *"As a manager, I want to log in securely so that I can view and manage my team's information, including leave management and other team-related tools."*

■ **Sub-User Stories:**

- As a manager, I want to securely log in with my unique credentials so that I can access my team's management tools without concerns about unauthorized access.

- As a manager, I want to be able to log out securely from the system after completing my tasks, ensuring no unauthorized access after I leave.

Admin Login and Authentication

■ **User Story:**

- *“As an admin, I want to log in securely so that I can manage system settings and user roles, ensuring the overall integrity and security of the application.”*

■ **Sub-User Stories:**

- As an admin, I want the login process to be quick and seamless so that I can efficiently manage system settings and oversee user activities.
- As an admin, I want the login system to have the ability to track and log my session details for future reference and security purposes.

HR Login and Authentication

■ **User Story**

- *“As an admin, I want to log in securely so that I can manage system settings and user roles, ensuring the overall integrity and security of the application.”*

■ **Sub-User Stories:**

- As an HR, I want to securely log in with my HR credentials so that I can access and manage employee records, leave requests, and other HR-related tasks.
- As an HR, I want to be able to log out securely after completing my tasks to ensure that no one else can access my HR management tools.

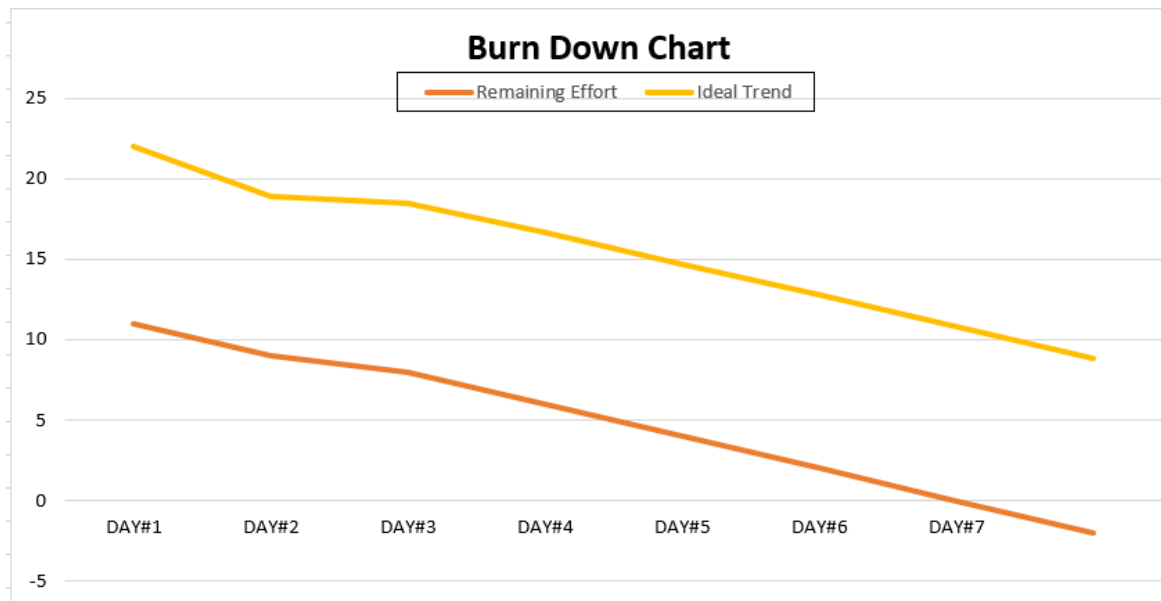
Product Backlog

| PRODUCT BACKLOG | | | | | | |
|-----------------|----------------|--|---|----------|--------|---------------|
| ID | AS A | I want to be able to | So that.... | Priority | Sprint | Status |
| 1 | Employee | request leave | I can take time off when needed | must | 1 | done |
| 2 | Manager | approve or reject leave requests | I can manage team availability | must | 1 | done |
| 3 | HR | process employee payroll | salaries are disbursed on time | must | 3 | to be started |
| 4 | Employee | view my salary slip | I can keep track of my earnings | must | 3 | to be started |
| 5 | Employee | view and update my profile | my information remains up to date | must | 1 | done |
| 6 | Manager | track employees' leave balances | I can decide if they are eligible for more leave | must | 3 | to be started |
| 7 | HR | add new employees to the system | their details are recorded for payroll and leave management | must | 1 | done |
| 8 | HR | remove employees from the system | records remain up to date and payroll and leave data are properly managed | must | 3 | to be started |
| 9 | Employee | check the approval status of my pending leave requests | I know whether my leave has been approved or rejected | must | 3 | to be started |
| 10 | HR | generate leave reports that summarize each employee's leave usage and averages | I can effectively monitor leave patterns and manage workforce planning | must | 3 | to be started |
| 11 | Admin | add new HR managers and department managers | they can manage employees effectively | must | 2 | in progress |
| 12 | Admin | remove HR managers and department managers | the system stays up-to date with active staff | must | 2 | in progress |
| 13 | Admin | view a list of all HR managers and department managers | I can track assigned personnel | must | 3 | to be started |
| 14 | Manager Login | log in to the system securely | I can access my personalized dashboard and role-specific functionalities | must | 2 | in progress |
| 15 | Admin Login | log in to the system securely | I can access my personalized dashboard and role-specific functionalities | must | 2 | in progress |
| 16 | HR Login | log in to the system securely | I can access my personalized dashboard and role-specific functionalities | must | 2 | in progress |
| 17 | Employee Login | log in to the system securely | I can access my personalized dashboard and role-specific functionalities | must | 2 | in progress |

Sprint Backlog

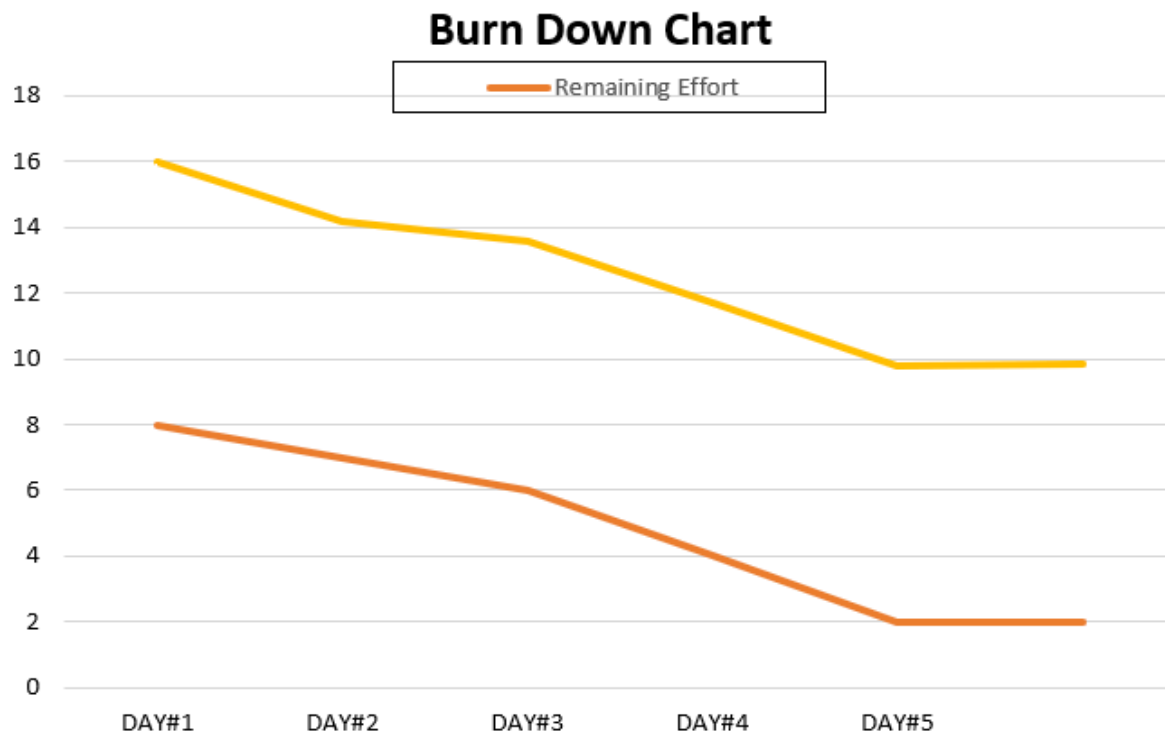
Sprint-01

| SPRINT BACKLOG | | | | | | | | | |
|----------------|-----------------------------|------------------|--------|--------|-------------|--------|--------|-------------|-------------|
| BACKLOG ID | USER STORIES | INITIAL ESTIMATE | 14-Feb | 15-Feb | 16-Feb | 17-Feb | 18-Feb | 19-Feb | 20-Feb |
| | | | DAY#1 | DAY#2 | DAY#3 | DAY#4 | DAY#5 | DAY#6 | DAY#7 |
| 1 | Employee Leave Request | 2 | 1 | | | 1 | | | |
| 2 | Leave Approval by Manager | 2 | | | 1 | | | 1 | 1 |
| 3 | Payroll Processing | 2 | | | | | 1 | | 1 |
| 4 | Employee Salary Slip Access | 3 | | 1 | | 1 | 1 | 1 | |
| 5 | View Employee Profile | 2 | 1 | | 1 | | | | |
| | Remaining Effort | 11 | 9 | 8 | 6 | 4 | 2 | 0 | -2 |
| | Ideal Trend | 11 | 9.9 | 10.45 | 10.63333333 | 10.725 | 10.78 | 10.81666667 | 10.84285714 |



Sprint-02

| SPRINT BACKLOG | | | | | | | |
|----------------|-----------------------------------|------------------|-------|-------|----------|-------|-------|
| BACKLOG ID | USER STORIES | INITIAL ESTIMATE | 2-Mar | 3-Mar | 4-Mar | 5-Mar | 6-Mar |
| | | | DAY#1 | DAY#2 | DAY#3 | DAY#4 | DAY#5 |
| 1 | Add HR and Managers | 2 | | 1 | | | |
| 2 | Remove HR and Managers | 2 | 1 | | | | |
| 3 | Employee Login and Authentication | 1 | | | | 1 | |
| 4 | Manager Login and Authentication | 1 | | | 1 | | |
| 5 | HR Login and Authentication | 1 | | | | 1 | |
| 6 | Admin Login and Authentication | 1 | | | 1 | | |
| | Remaining Effort | 8 | 7 | 6 | 4 | 2 | 2 |
| | Ideal Trend | 8 | 7.2 | 7.6 | 7.733333 | 7.8 | 7.84 |



Project Plan

Project Initiation

- **Define scope:** What PayTrack will do (employee salary management, manager approvals, admin controls, etc.)
- **Stakeholders:** users - admin, employees, managers, hr
- **Requirements gathering:**
 - User roles: Admin, Manager, Employee

- Main features: Employee management, Payroll processing, Leave management, Reports

Architecture & Setup

- **Decide tech stack:**
 - Backend: Spring Boot (Java), MySQL
 - Frontend: ReactJS, Tailwind or plain CSS
- **Set up repos:** GitHub
- **Create base skeleton:**
 - Spring Boot initial project (dependencies: Spring Web, Spring Data JPA, MySQL Driver, Spring Security if needed)
 - React project (create-react-app)

Backend Development (Spring Boot)

Phase 1: Models + Repositories

- Entities: Employee, Manager, Admin, Leaves, HR, Manager
- Repositories (interfaces for DB access)

Phase 2: Services

- Business logic:
 - Add employee
 - Payroll processing
 - Approve leave
 - Leave Reports

Phase 3: Controllers (APIs)

- RESTful endpoints:
 - /employees
 - /users
 - /leaves
 - /admins
 - /hrs
 - /managers
 - /departments

Frontend Development (ReactJS)

Phase 1: Setup Pages

- Create individual folders for user interfaces eg Administrator, employee, HR and Manager

Phase 2: Components

- Layout
- Sidebar
- Cards (for employee leave requests)

Phase 3: API Integration

- Fetch data from backend

- Post data (like submitting leave requests, adding employees)

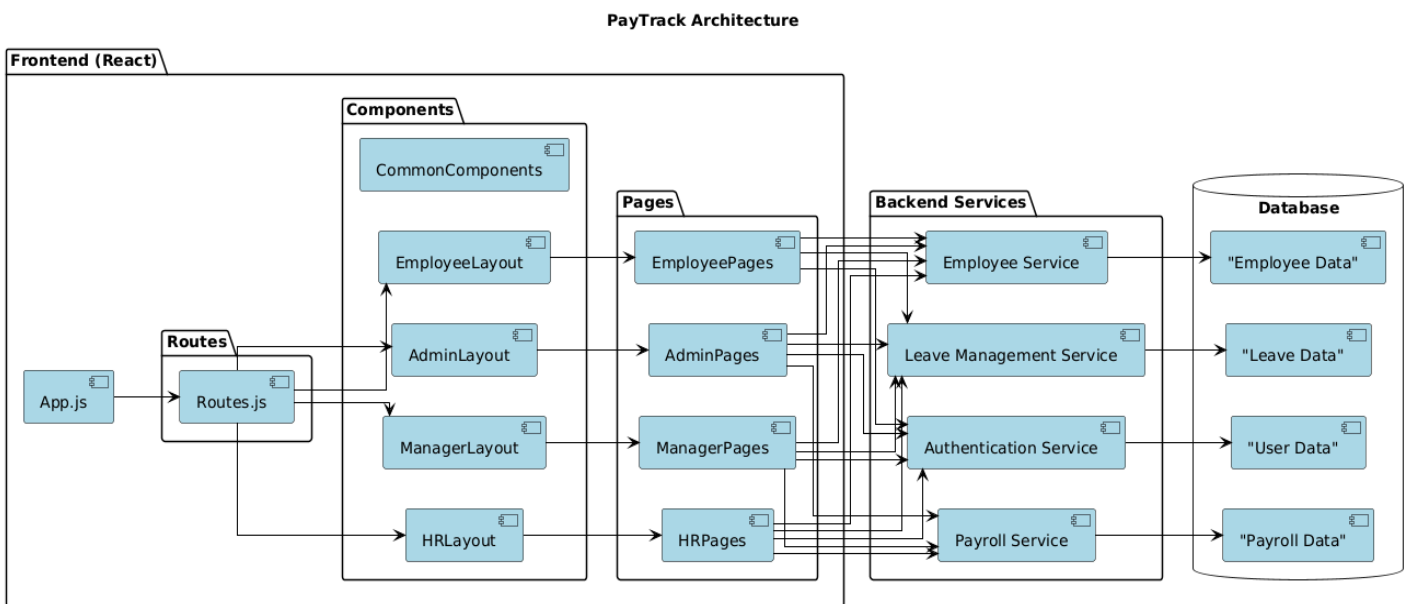
Testing

- Backend:
 - Unit tests (Spring Boot JUnit tests for services)
- Frontend:
 - Manual testing: Form submissions, table updates, API responses (idk didn't do yet)

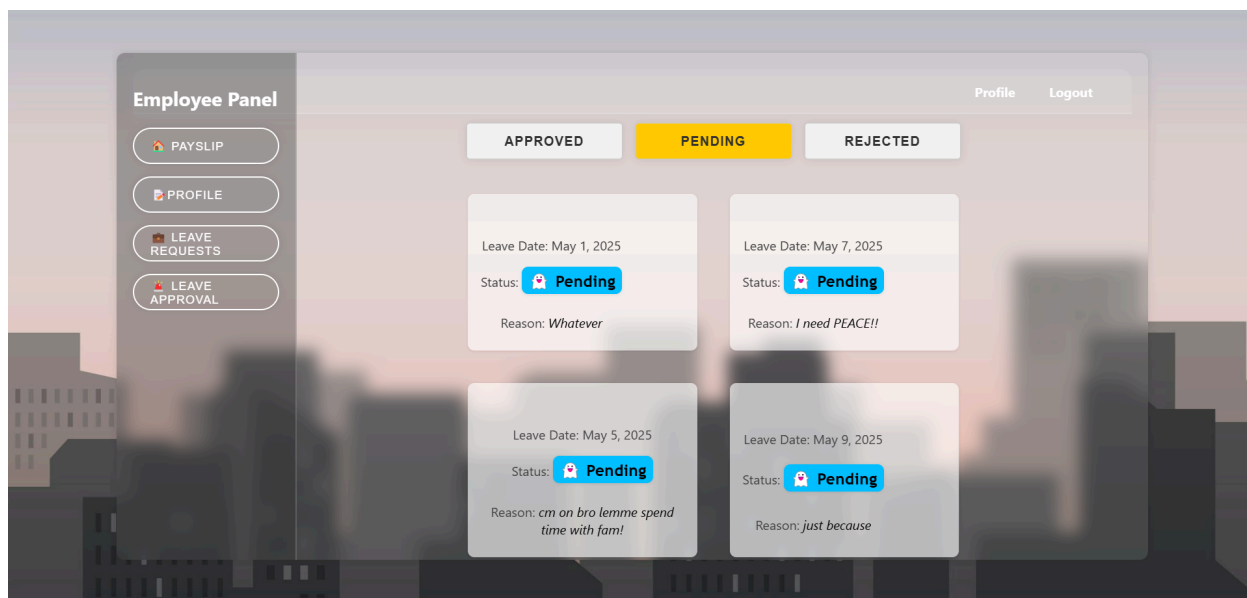
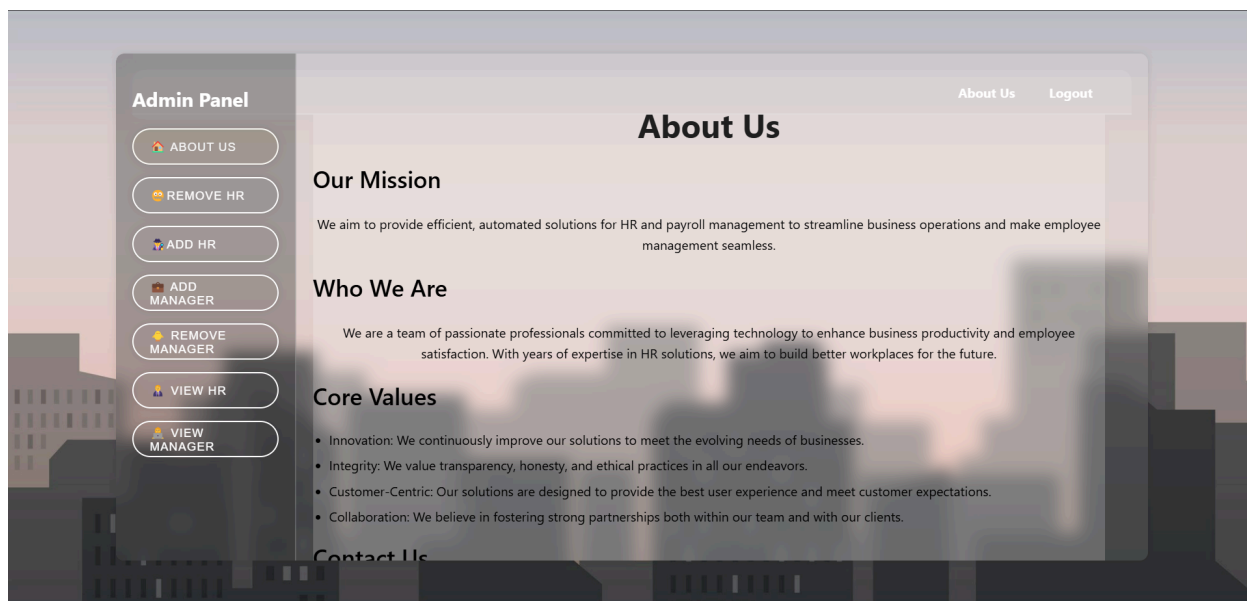
Final Deliverables

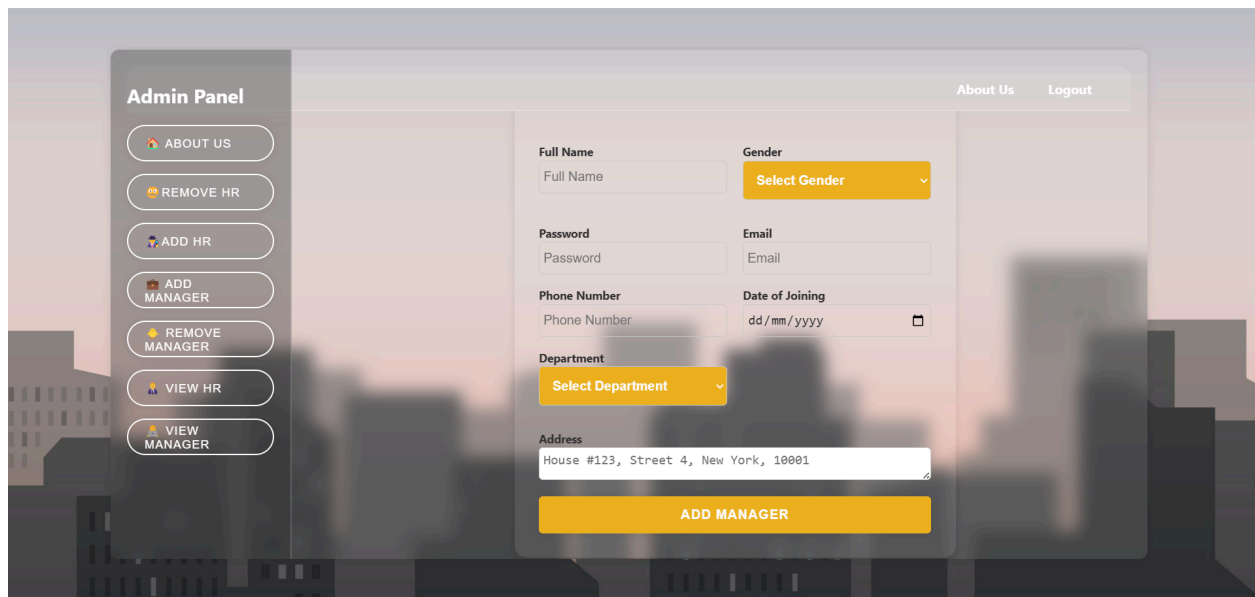
- Project report / documentation
- GitHub repo

Architecture Diagram



Actual implementation Screenshot





The image shows a web application interface for an Admin Panel. On the left, there is a vertical sidebar with the title "Admin Panel" and a list of buttons: "ABOUT US", "REMOVE HR", "ADD HR", "ADD MANAGER", "REMOVE MANAGER", "VIEW HR", and "VIEW MANAGER". The main content area is titled "About Us" and "Logout" in the top right corner. It contains a form for adding a new manager. The form fields are: "Full Name" (text input), "Gender" (dropdown menu with "Select Gender" as the placeholder), "Password" (text input), "Email" (text input), "Phone Number" (text input), "Date of Joining" (calendar icon), "Department" (dropdown menu with "Select Department" as the placeholder), and "Address" (text input with the value "House #123, Street 4, New York, 10001"). A large yellow button labeled "ADD MANAGER" is at the bottom of the form.

Admin Panel

ABOUT US

REMOVE HR

ADD HR

ADD MANAGER

REMOVE MANAGER

VIEW HR

VIEW MANAGER

About Us Logout

Full Name

Full Name

Gender

Select Gender

Password

Password

Email

Email

Phone Number

Phone Number

Date of Joining

dd/mm/yyyy

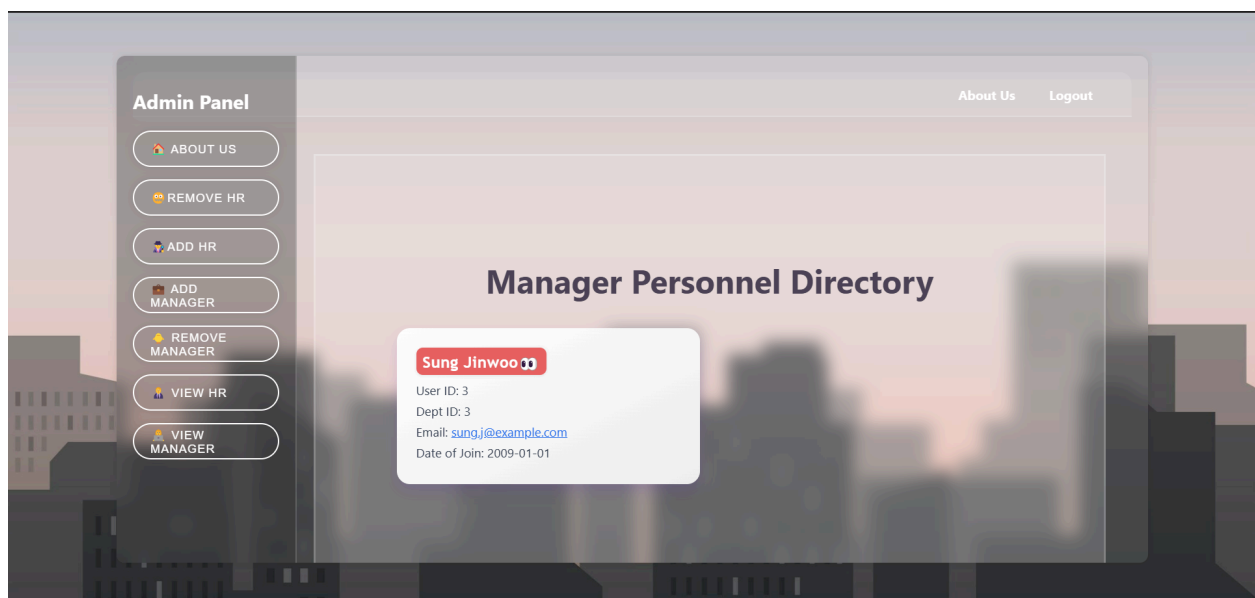
Department

Select Department

Address

House #123, Street 4, New York, 10001

ADD MANAGER



The image shows the same Admin Panel interface, but the main content area now displays the "Manager Personnel Directory". It features a large heading "Manager Personnel Directory" and a card for a specific manager. The card displays the name "Sung Jinwoo" in a red box, followed by his details: "User ID: 3", "Dept ID: 3", "Email: sungj@example.com", and "Date of Join: 2009-01-01". The sidebar and top navigation remain the same.

Admin Panel

ABOUT US

REMOVE HR

ADD HR

ADD MANAGER

REMOVE MANAGER

VIEW HR

VIEW MANAGER

About Us Logout

Manager Personnel Directory

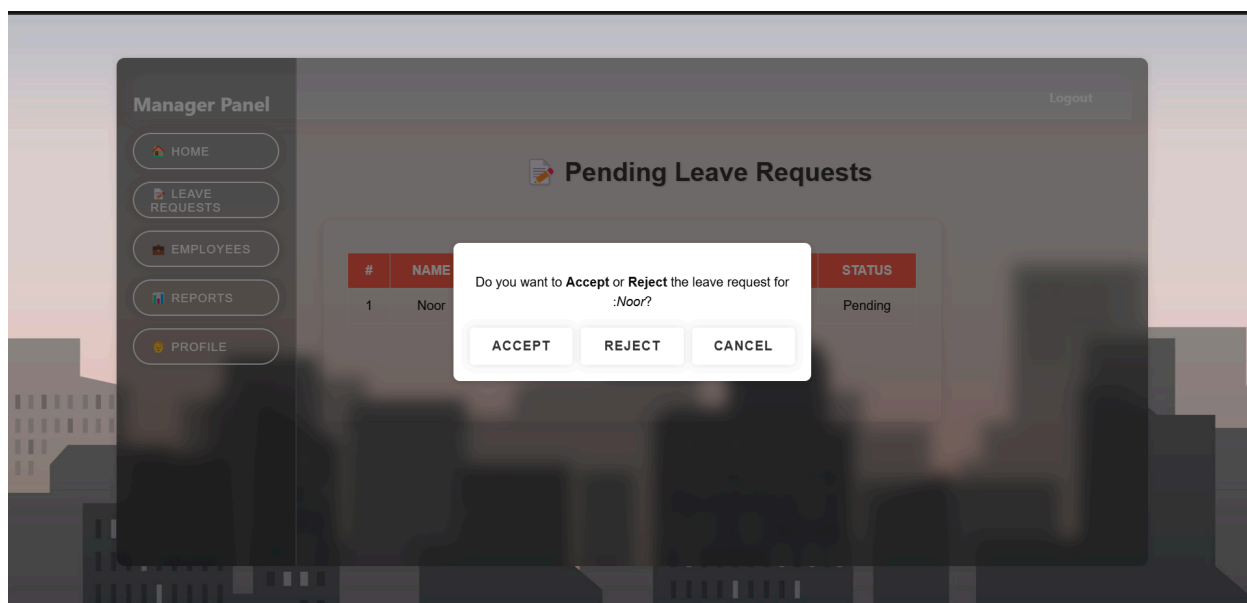
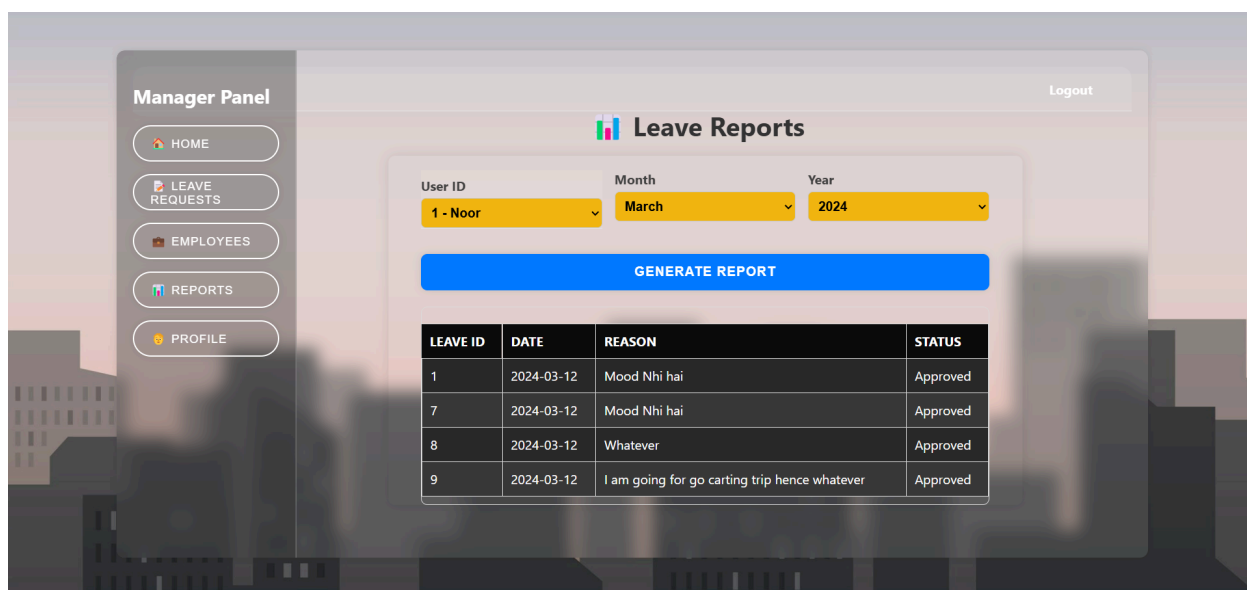
Sung Jinwoo

User ID: 3

Dept ID: 3

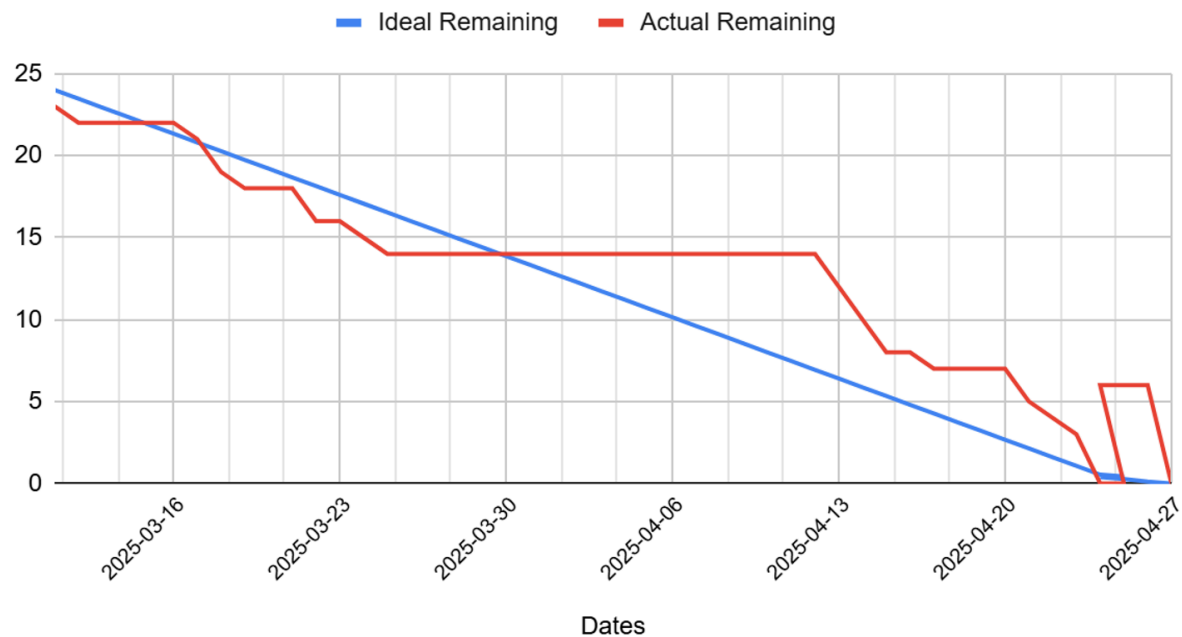
Email: sungj@example.com

Date of Join: 2009-01-01



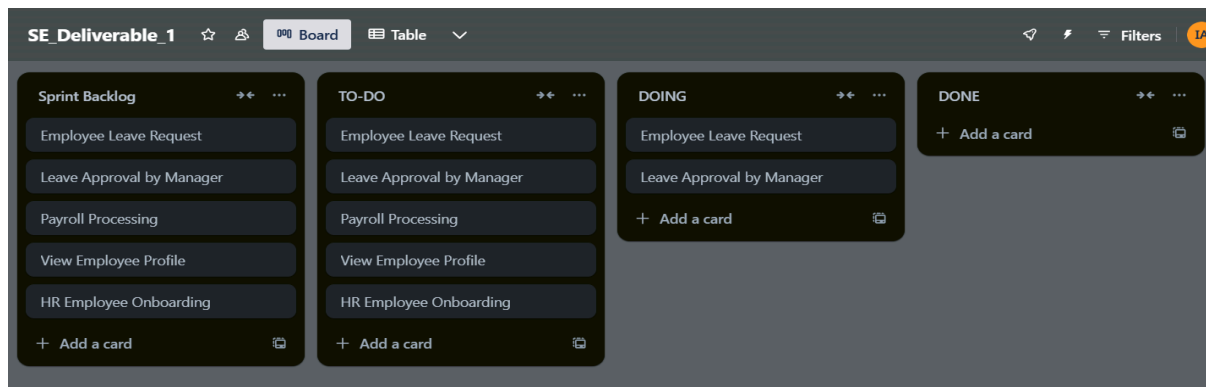
Product burndown chart (complete project)

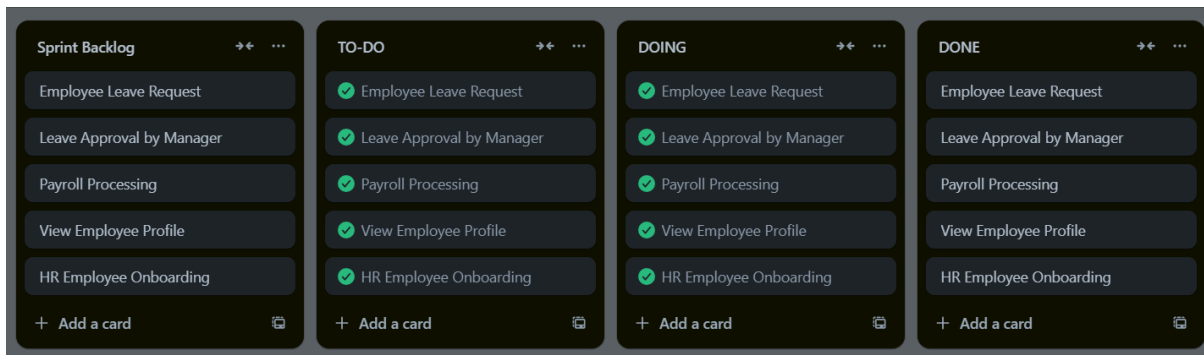
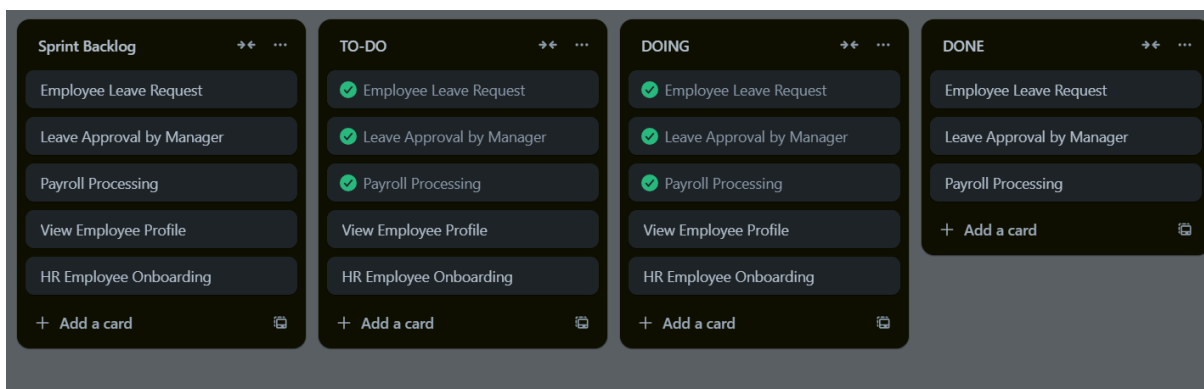
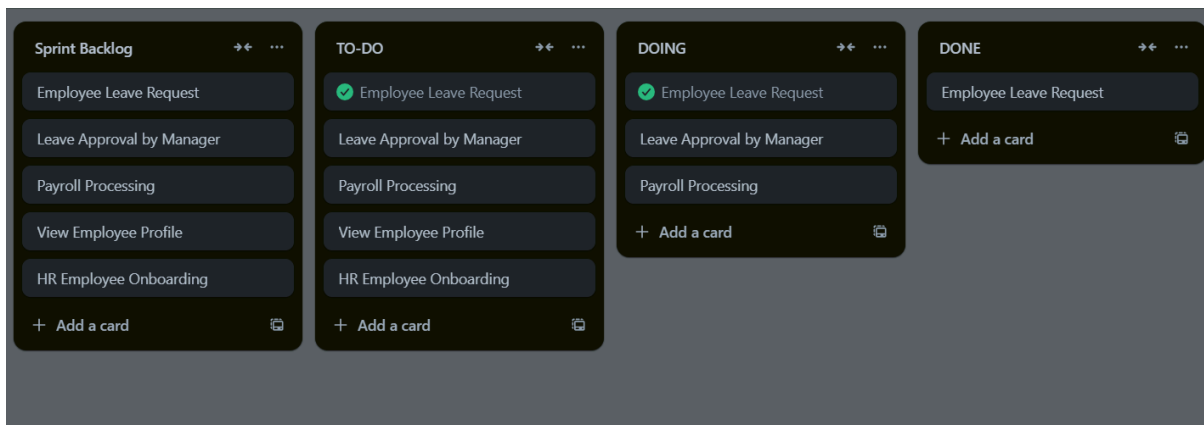
Project BurnDown Chart



Trello Board Screenshots

Sprint-01





Sprint-02

SE_Deliverable_2 ☆ Board

Sprint Backlog

- Add HR and Manager
- Remove HR and Manager
- Employee Login and Authentication
- Manager Login and Authentication
- Admin Login and Authentication
- HR Login and Authentication
- + Add a card

TO-DO

- Add HR and Manager
- Remove HR and Manager
- Employee Login and Authentication
- Manager Login and Authentication
- Admin Login and Authentication
- HR Login and Authentication
- + Add a card

DOING

- Add HR and Manager
- Remove HR and Manager
- Employee Login and Authentication
- + Add a card

DONE

- + Add a card

SE_Deliverable_2 ☆ Board

Sprint Backlog

- Add HR and Manager
- ☐ Remove HR and Manager
- Employee Login and Authentication
- Manager Login and Authentication
- Admin Login and Authentication
- HR Login and Authentication
- + Add a card

TO-DO

- Add HR and Manager
- Remove HR and Manager
- Employee Login and Authentication
- Manager Login and Authentication
- Admin Login and Authentication
- HR Login and Authentication
- + Add a card

DOING

- ✓ Add HR and Manager
- ✓ Remove HR and Manager
- ✓ Employee Login and Authentication
- Manager Login and Authentication
- Admin Login and Authentication
- HR Login and Authentication
- + Add a card

DONE

- Add HR and Manager
- Remove HR and Manager
- Employee Login and Authentication
- + Add a card

SE_Deliverable_2 ☆ Board

Sprint Backlog

- Add HR and Manager
- Remove HR and Manager
- Employee Login and Authentication
- Manager Login and Authentication
- Admin Login and Authentication
- HR Login and Authentication
- + Add a card

TO-DO

- Add HR and Manager
- Remove HR and Manager
- Employee Login and Authentication
- Manager Login and Authentication
- Admin Login and Authentication
- HR Login and Authentication
- + Add a card

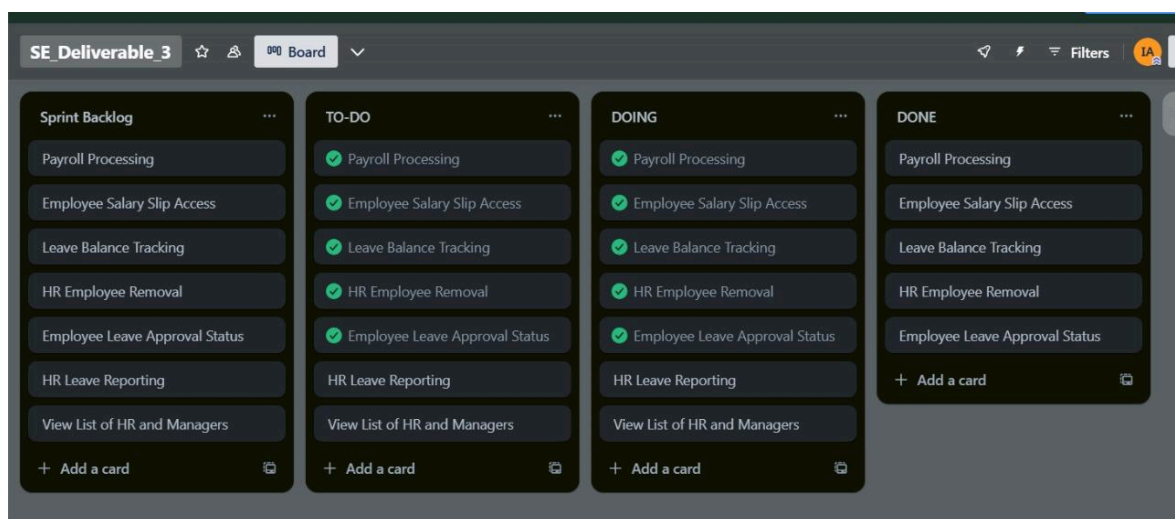
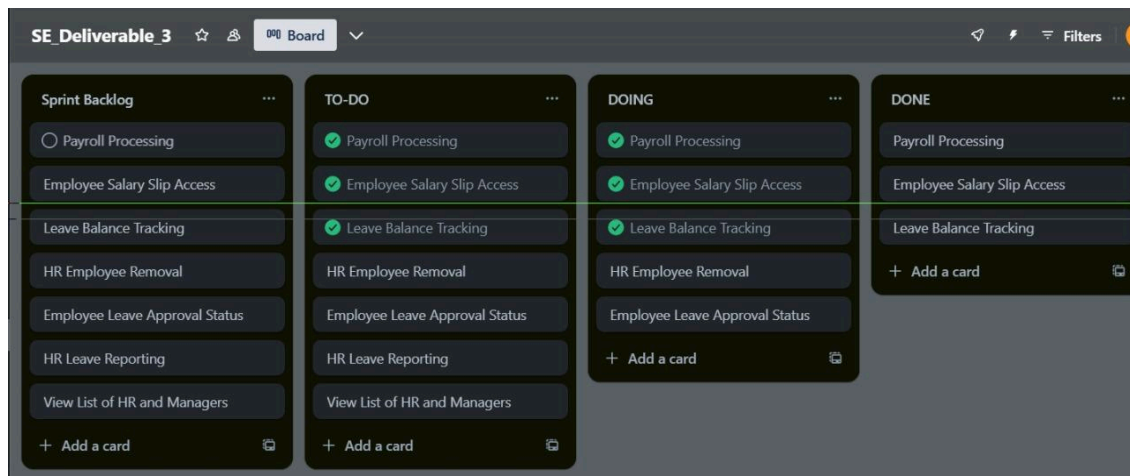
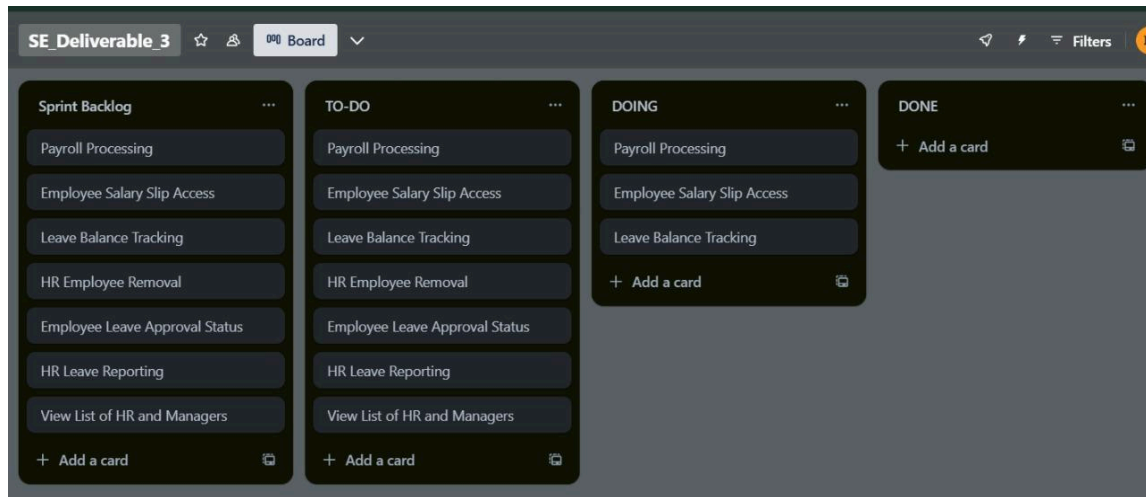
DOING

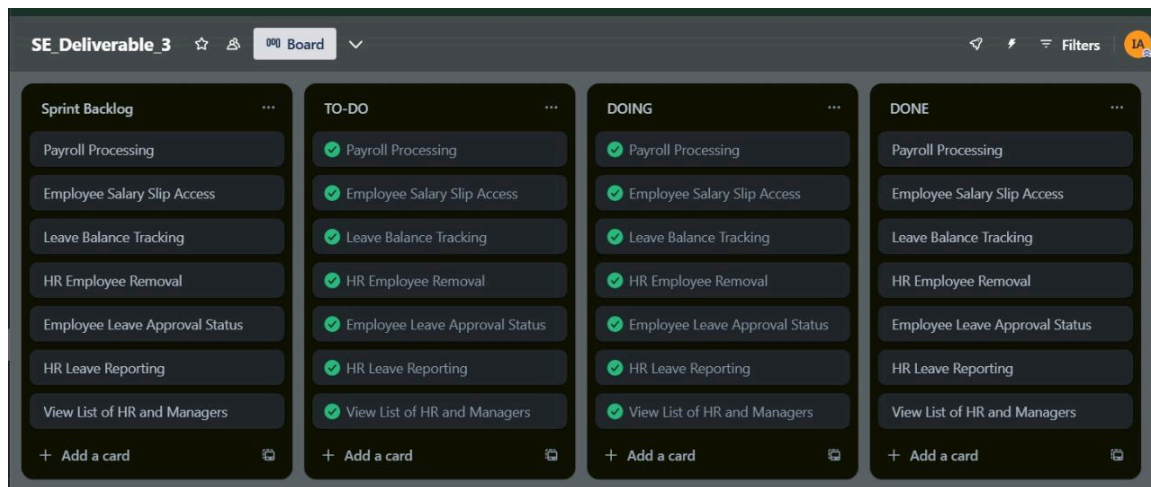
- ✓ Add HR and Manager
- ✓ Remove HR and Manager
- ✓ Employee Login and Authentication
- ✓ Manager Login and Authentication
- ✓ Admin Login and Authentication
- ✓ HR Login and Authentication
- + Add a card

DONE

- Add HR and Manager
- Remove HR and Manager
- Employee Login and Authentication
- Manager Login and Authentication
- Admin Login and Authentication
- HR Login and Authentication
- + Add a card

Sprint-03





Test Cases Blackbox

| Test Case ID | Description | Inputs | Expected Output | Actual Output | Type | Status |
|--------------|---|--------------------------------|-------------------------------------|---|--------------------------|--------|
| TC01 | Verify login with valid credentials | Username: Noor, Password: n122 | Redirect to Employee Profile | Successfully redirected to profile page | Equivalence Partitioning | Pass |
| TC02 | Verify login with empty username | Username: "", Password: 123456 | Error: Please fill out this field | Doesn't redirect to profile page, pops a help message | Equivalence Partitioning | Fail |
| TC03 | Add Employee with valid details | Name: John Doe, Salary: 50000 | Employee added successfully | Shows an alert message of adding the employee | Equivalence Partitioning | Pass |
| TC04 | Add Employee with missing salary | Name: Jane Doe, Salary: "" | Error: Salary is required | Fails to add employee | Equivalence Partitioning | Fail |
| TC05 | Add Employee with incorrect email (missing @) | Email: noor.com | Error: Invalid format alert message | Alert message asking the user to add @ | Equivalence Partitioning | Fail |

| | | | | | | |
|------|---|--------------------------------|--|--|---------------------------------|------|
| TC06 | Adding employee with many missing fields of input | Salary: Phone number: | Error message and alert mentioning to add these fields | Alert message to add values in these fields | Equivalence Partitioning | Fail |
| TC07 | Generate payslip for active employee | EmployeeID: 123 | Payslip generated | Payslip generated | Equivalence Partitioning | Pass |
| TC08 | Generate payslip for invalid employee | EmployeeID: 999 | Error: Employee not found | Can't generate for employee who is not in DB | Equivalence Partitioning | Fail |
| TC09 | View salary details when logged in | Username: Noor, Password: n122 | Display salary details | Displays salary after any deductions | Equivalence Partitioning | Pass |
| TC10 | Manager approves valid pending leave request | PaymentID: 456 | Payment status updated to "Approved" | Changes status of the leave request | Equivalence Partitioning | Pass |

Test Cases whitebox

Backend

demo

demo

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes | |
|---|------------------------|--------------|------------------------|-----------|--------|------|--------|-------|--------|---------|--------|---------|----|
| com.example.demo.controller | <div><div></div></div> | 76% | <div><div></div></div> | 78% | 34 | 115 | 56 | 244 | 19 | 70 | 1 | 8 | |
| com.example.demo.service | <div><div></div></div> | 74% | <div><div></div></div> | 77% | 22 | 71 | 32 | 133 | 16 | 53 | 0 | 7 | |
| com.example.demo.model | <div><div></div></div> | 86% | | n/a | 8 | 61 | 13 | 88 | 8 | 61 | 0 | 7 | |
| com.example.demo | | 37% | | n/a | 1 | 2 | 2 | 3 | 1 | 2 | 0 | 1 | |
| Total | | 418 of 1,826 | 77% | 27 of 126 | 78% | 65 | 249 | 103 | 468 | 44 | 186 | 1 | 23 |

The backend white box testing was performed using **JUnit** for unit testing and **JaCoCo** for generating the coverage report. The testing focused on verifying all key functions and methods across the controller, service, and model layers. The coverage report achieved:

- **Statement Coverage (lines of code): 77%**
- **Branch Coverage (if/else paths): 78%**

- **Function/Method Coverage:** Above 70% across all major layers

What is Covered Well

- Core business logic in **major services** (Admin, User, Manager, Employee) was thoroughly tested.
- API endpoints of important controllers like **AdminController**, **EmployeeController**, and **ManagerController** are well covered.
- Model classes and their basic operations were adequately tested.

What is Not Covered and Why

- Some controllers (UserController, DepartmentController) and services (LeaveService, DepartmentService) were not fully covered, primarily due to missing specific unit tests for their functionalities.
- Certain branches inside HRService were not triggered during unit tests because they require rare or error-prone input scenarios.
- External dependencies and third-party libraries are outside the project's direct testing scope.

Frontend

For the PayTrack project, I used **Jest** to test the frontend React components. Most components achieved high coverage, especially in the src/components folder. The

| File | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s |
|--------------------------------|--------------|-------------|--------------|--------------|-------------------|
| All files | 8.69 | 2.72 | 16.53 | 8.95 | |
| src | 9.09 | 0 | 33.33 | 9.09 | |
| App.js | 100 | 100 | 100 | 100 | |
| index.js | 0 | 100 | 100 | 0 | 8-9 |
| reportWebVitals.js | 0 | 0 | 0 | 0 | 1-8 |
| src/components | 91.89 | 70 | 93.02 | 91.89 | |
| AdminLayout.js | 100 | 100 | 100 | 100 | |
| AdminSidebar.js | 100 | 100 | 100 | 100 | |
| EmployeeLayout.js | 100 | 100 | 100 | 100 | |
| EmployeeSidebar.js | 100 | 100 | 100 | 100 | |
| HRLayout.js | 100 | 100 | 100 | 100 | |
| HRSidebar.js | 100 | 100 | 100 | 100 | |
| LeaveRequestCard.js | 87.5 | 87.5 | 100 | 87.5 | 15 |
| LeaveRequestTab.js | 0 | 0 | 0 | 0 | 5-12 |
| ManagerLayout.js | 60 | 100 | 50 | 60 | 36-37 |
| ManagerSidebar.js | 100 | 100 | 100 | 100 | |
| badge.js | 100 | 100 | 100 | 100 | |
| card.js | 100 | 100 | 100 | 100 | |
| tabs.js | 100 | 100 | 100 | 100 | |
| src/pages/Administrator | 0.8 | 0 | 1.31 | 0.82 | |
| AboutUS.js | 100 | 100 | 100 | 100 | |

Software Engineering Project Work Division

Our team collaboratively designed and implemented a comprehensive software system by leveraging individual strengths and interests. The work was strategically divided across sprints to ensure consistency, efficiency, and seamless integration. Below is the detailed division of responsibilities:

Sprint 1: Frontend Development

- **Frontend Development:** Noor Fatima developed the entire frontend, including all ReactJS components, ensuring a consistent design and functionality across the project, except for the Manager Interface.
- **Manager Interface:** Munim fully developed the Manager Interface, focusing on its unique frontend requirements.

Module Development

- **Employee Interface:** Entirely developed by Noor Fatima, encompassing all frontend and backend functionalities.
- **Administrator Interface:** 90% developed by Sara, including the majority of interface design and integration, with a few specific functions implemented by Noor Fatima.
- **Manager Interface:** Fully handled by Munim, covering both frontend and backend development.
- **HR Interface:** Equally divided among Noor Fatima, Sara, and Munim, with each contributing to both frontend and backend tasks based on their prior sprint responsibilities.
- **Login Functionalities:**
 - All login functionalities, except for the Administrator login, were developed by Munim.
 - The Administrator login was integrated as part of the Administrator Interface by Sara.

Backend Development

- Based on assigned user stories, each team member completed their respective backend tasks, ensuring smooth integration with the frontend modules.

User Stories and Task Management

- **User Stories:** Created by Munim to define clear functional and non-functional requirements for development.
- **Trello Work:** All Trello board organization, burndown charts and task tracking were managed by Sara to ensure smooth workflow and team coordination.

Documentation

- **Package Diagram and Component Diagram:** Created by Sara.
- **Gantt Chart and Work Breakdown Structure:** Developed by Noor Fatima.
- **Architecture and Deployment Diagram:** Designed by Munim.
- **Presentation (PPT):** Prepared by Sara.
- **Final Report:** Written by Noor Fatima.

Testing

- **Whitebox and Blackbox Testing:** Equally divided among Noor Fatima, Sara, and Munim, with each focusing primarily on the interfaces they developed in previous sprints to ensure thorough validation.

Lessons Learned from Software Engineering Project

Our team's collaborative effort in designing and implementing the software system, supported by effective version control practices, provided valuable insights into software engineering. We utilized GitHub for version control, maintaining three distinct branches to manage development workflows. Below are the key lessons learned based on the work division and project execution:

Key Lessons Learned

- **Effective Task Division Enhances Output:** Assigning tasks based on individual strengths streamlined development and improved quality.
- **Consistency Reduces Rework:** Having one developer handle most frontend work ensured a uniform design, minimizing later refactoring.
- **Clear Communication Prevents Overlaps:** Dividing shared modules emphasized the importance of regular coordination and clear task boundaries.
- **Version Control Supports Collaboration:** Using GitHub with structured branching enabled smooth concurrent development, though resolving conflicts required careful teamwork.
- **Balanced Documentation Strengthens Clarity:** Distributing documentation responsibilities ensured comprehensive and organized project records.
- **Focused Testing Improves Quality:** Aligning testing responsibilities with development ownership led to better defect detection and deeper module understanding.
- **Iterative Feedback Enhances Teamwork:** Regular pull requests, code reviews, and check-ins helped identify issues early and fostered cohesive collaboration.
- **Time Management Ensures Timely Delivery:** Setting internal milestones and tracking progress helped maintain deadlines and project flow.

These experiences reinforced the value of skill-based task allocation, strong communication, robust version control practices, and continuous iterative improvement for successful project delivery.