DOUGLAS COLLEGE

Computing Studies & Information Systems

"Applied Research Project"

Fall 2022: CSIS 4495 – Section 050

Progress Report 2

"TeamSavvy"

by:

- **Name:** Jasmine Kaur **Student ID:** 300345933
- **Name:** Jaspal Singh **Student ID:** 300345938
- **Name:** Harman Lal **Student ID:** 300343291
- **Name:** Joling Weng **Student ID:** 300335548

DOUGLAS COLLEGE

Statement of the Problem:

The Employee Management System helps create an environment that motivates communication and keeps employees and managers working together to meet company goals. It also helps the company to manage the information and activities of employees.

The issue with the existing system is to access data about employee projects, leaves, payroll, and jobs applied internally, for accessing these details an individual has to use different web platforms.

Our system proposes that a person can access the details for projects, work history, team hierarchy, peers, timesheets, leaves and payroll information on one platform. The managers and company owners can access the records of employees and also check the resume of the candidates who applied for jobs on their company website with an improved user interface for a better user experience.

Significance of the study

Who would benefit from this project and how?
Employee Engagement: Our platform will provide employees to add their details, apply for leaves, and check their work history and payroll details on one platform. This will save time for an individual to concentrate on tasks for company growth. Employees can track their growth and improve their working efficiency.

Biometric: We will create an android app to get user biometric timestamp and will store it in the database. Employee can use his biometric to clock-in and clock-out of the system.

Create Dashboard: Our project will provide the dashboard of collective data of employees. A manager or company owner can perform descriptive analysis on employee's data; with respect to programming languages, projects, leaves (sick leave, casual leave) in specific period and, salary. **We will give different graph options to select from and, can create multiple graphs to visualize data and show the dashboard on preview mode and can download it.** The HR at a company can do analytics; for example, number of employees have knowledge of programming language, salary of each employee in specific project. We will represent this information in a graphical format.

Create Task: The employee can fill his hours he is working on a particular task and then change the status of task to in-progress, after completing the task employee can change the task status to finish. The notification will be sent to the manager that this task has been completed. Manager can track the progress of the tasks assigned to team members.

Improve communication: Our platform provides the information of each employee to others by search option that will let them to stay connected, and an easy path for new joiners to get to know their peers. All in all, improve the connection which will reduce the problem of communication gap within an organization.

Users:

Employee**:**

- An employee will have access to a calendar where he can see clock-in and clock-out for a day
- Check total working hours
- Apply for leaves
- Check his payroll details (Read only),
- Get information of other employee working in an organization for communication
- Can get the team hierarchy
- Check the number and details of projects he is working on
- Employee will have the profile page where he will have his information saved or can update if required
- Can apply on internal jobs
- Apply for resignation

Admin (HR or Manager):

- An admin will have all the functionality that employee has
- Able to see information of employees in his team
- He can perform CRUD operations on them
- Check payroll info and can give increment to an employee
- Admin will have details of project's budget and deadline
- Can assign project to employee
- Modify involvement of employees in different projects,
- Can approve leaves of an employee
- Check timesheets
- Create internal job postings for employees.

Super admin (Company Owner): A super admin can perform CRUD operations on admins and employees.

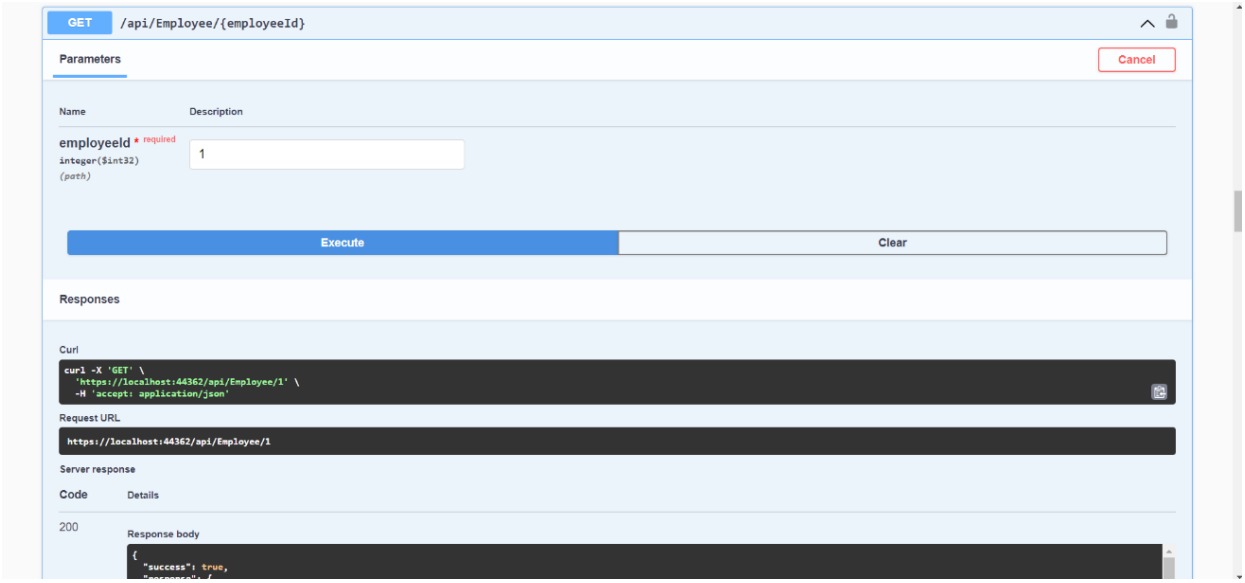
Project Implementations:

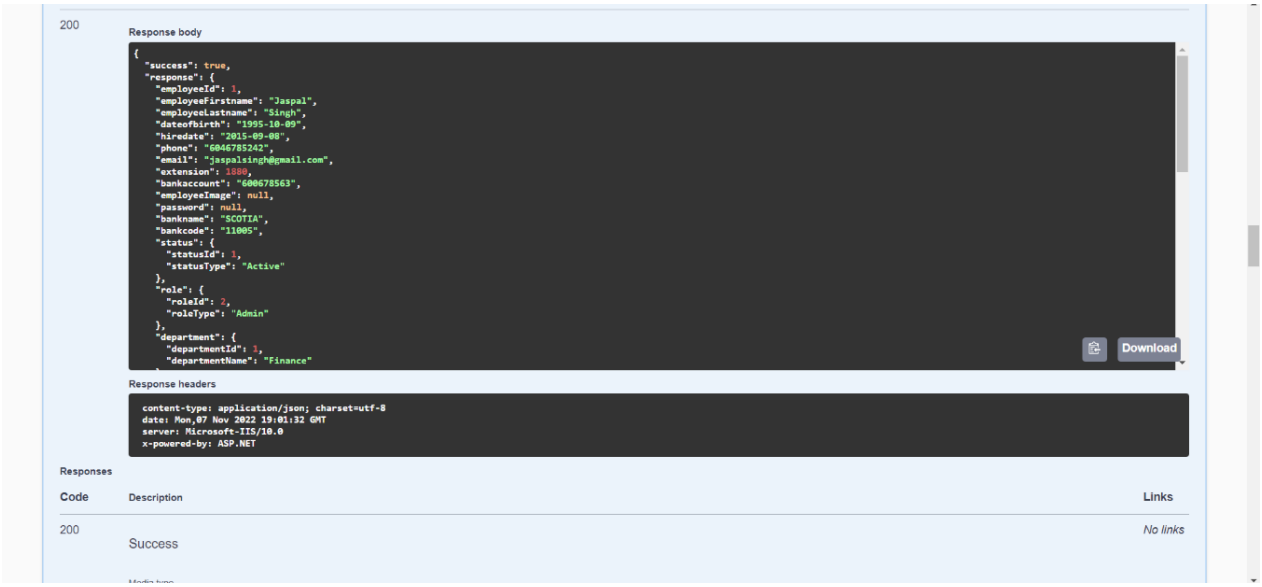
What have we done so far ?

Test Cases and screenshot: Testing the API gives the following results:

Checking the Employee Database:

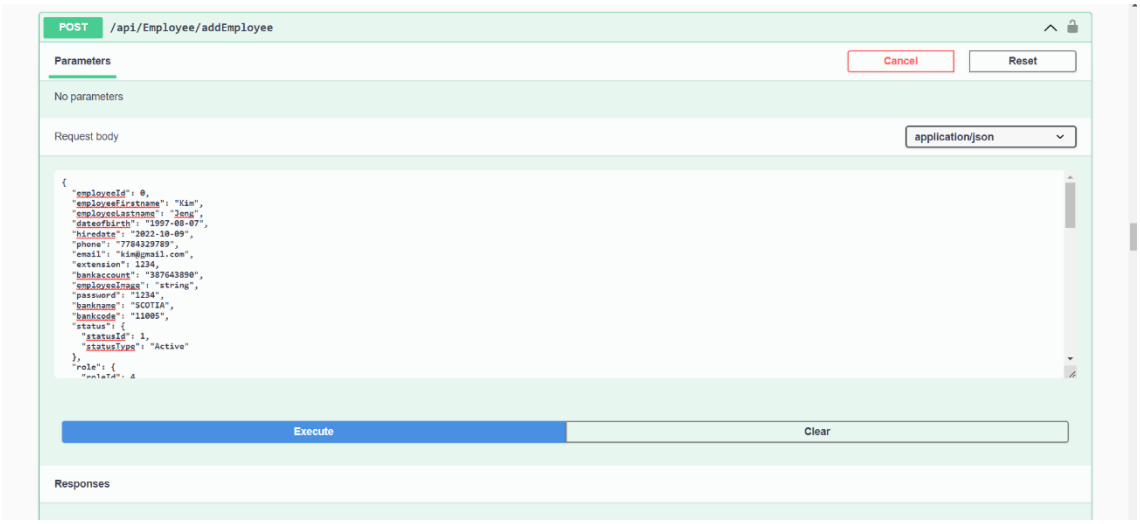
In this we enter the employee id which will correspond to the employee details associated to the database. As an output of this we get the response from the API with desired user details.

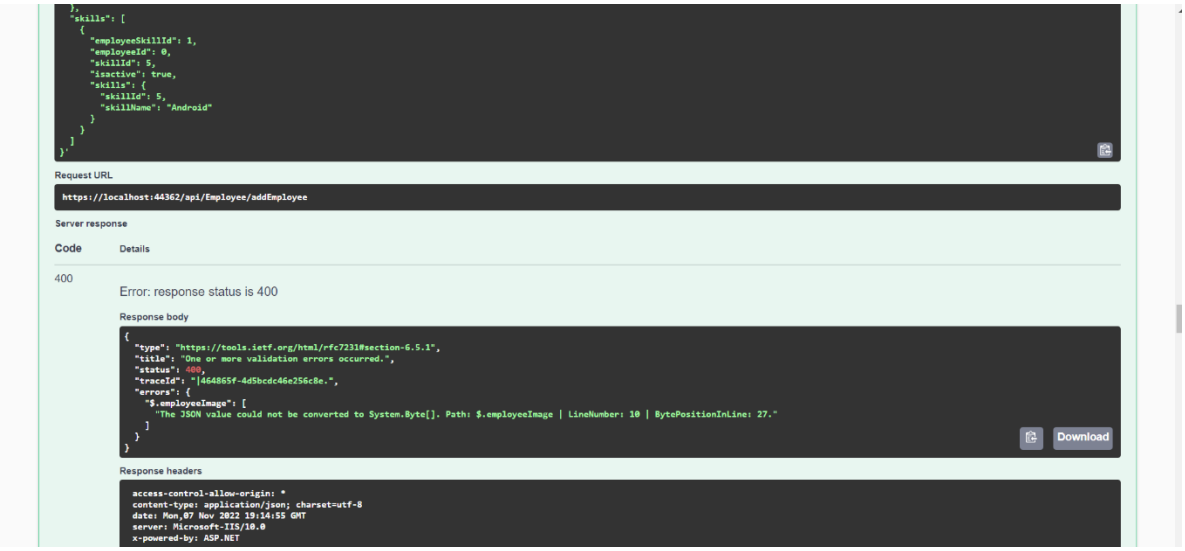




To add a new Employee :

When HR enter the details of a new employee. The API will save the data from frontend to database. This part of the API needs fixation as so far we're not able to add the Employee. It will generate an error stating that response status is 400.

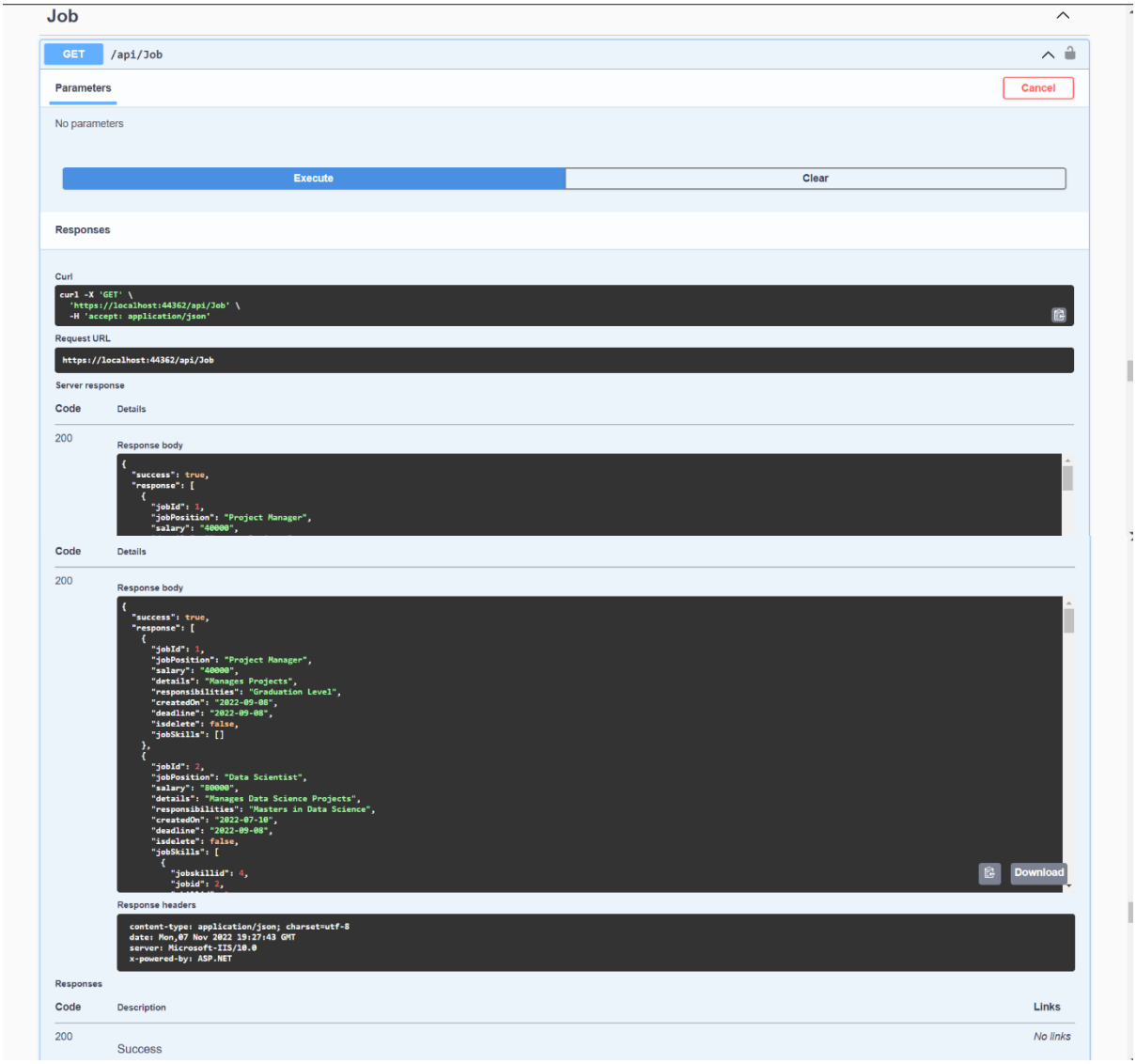




Due to this **Deletion of an Employee** cannot be done.

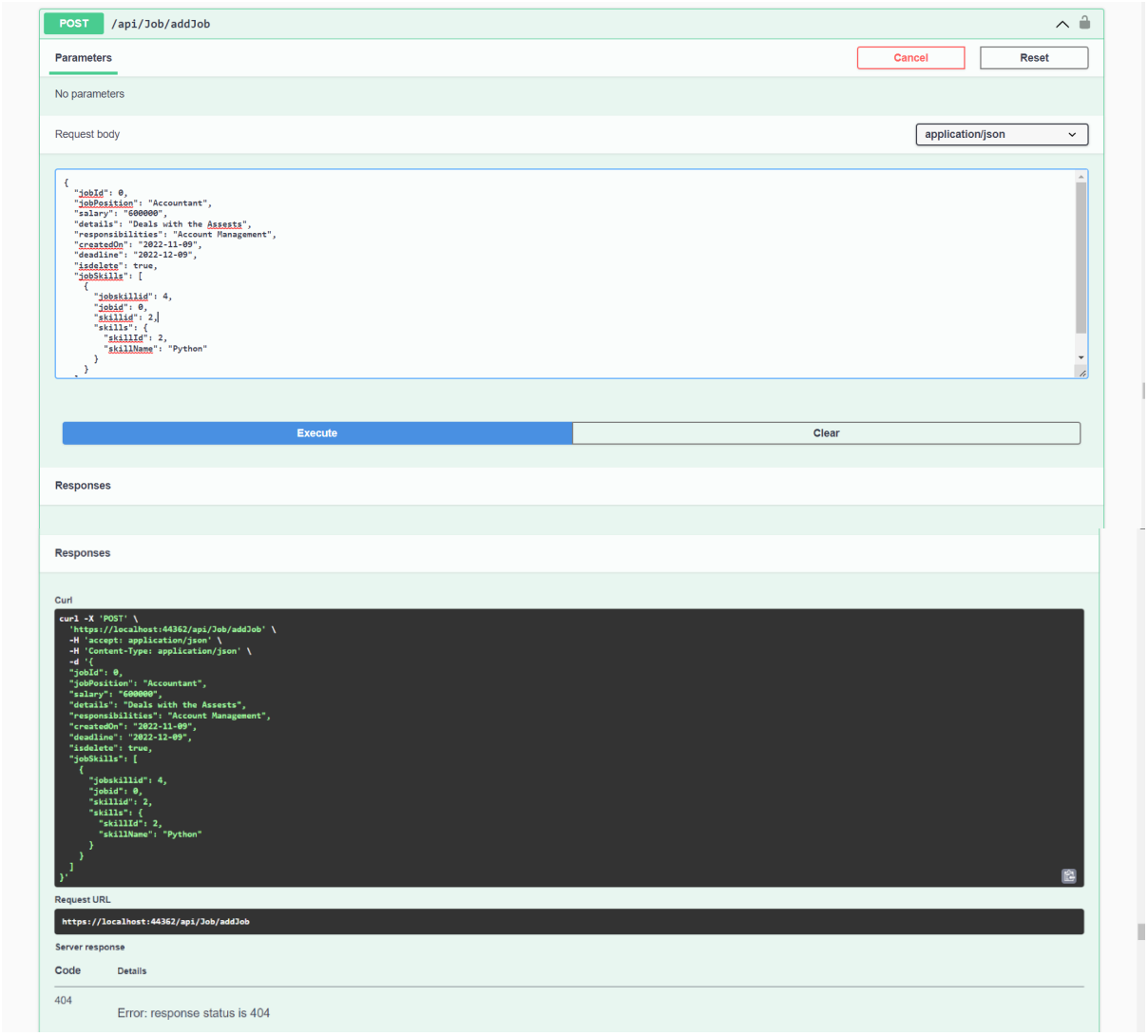
Jobs :

The job API retrieves the data from database that is related to the Jobs posted in the company.



Add a Job:

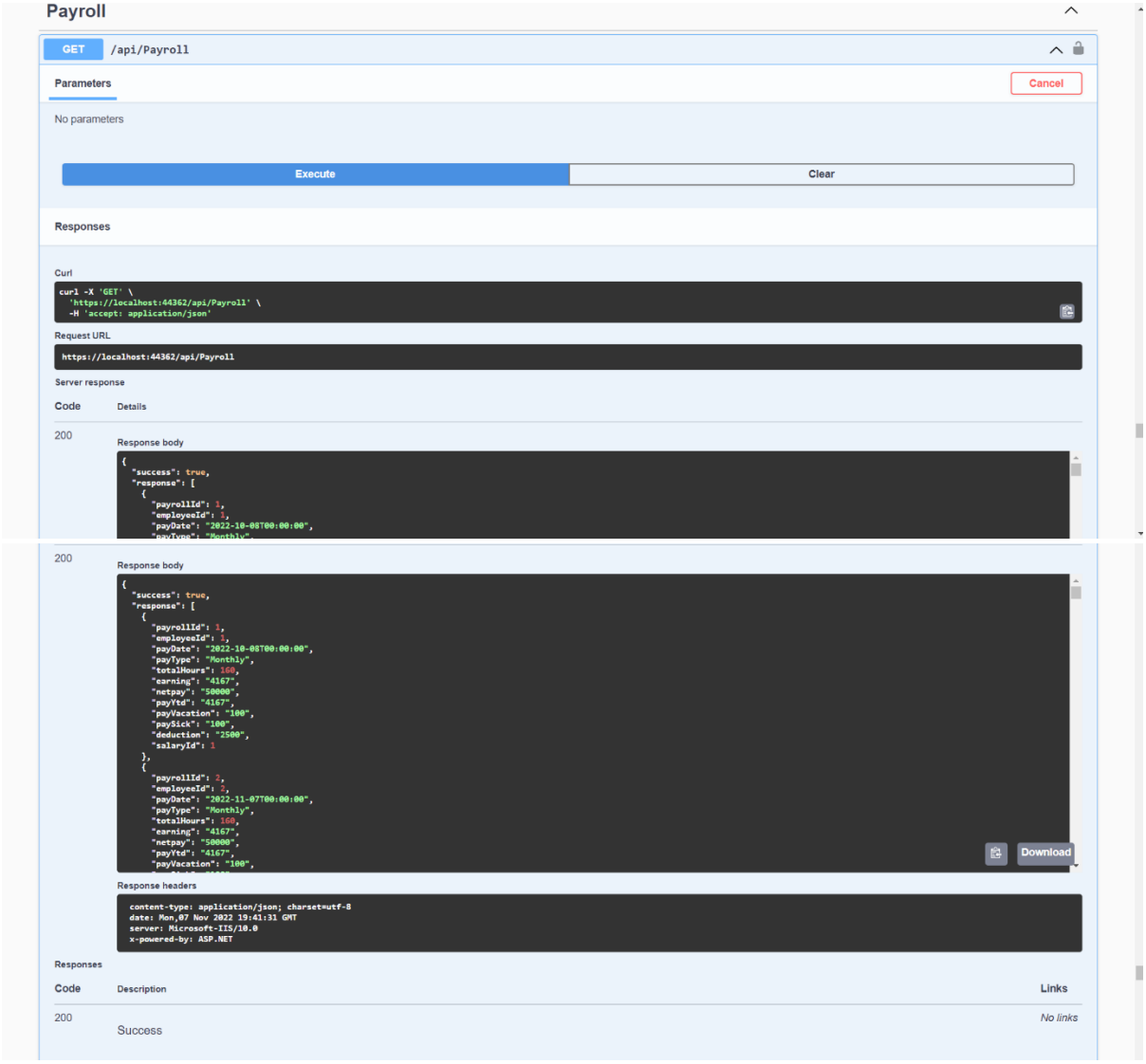
While adding a new job, generate a new job posting with required attributes. Simliar to adding the Employee, this part gives an error code 400 as well.



Therefore, the deletion of a job will not proceed as well.

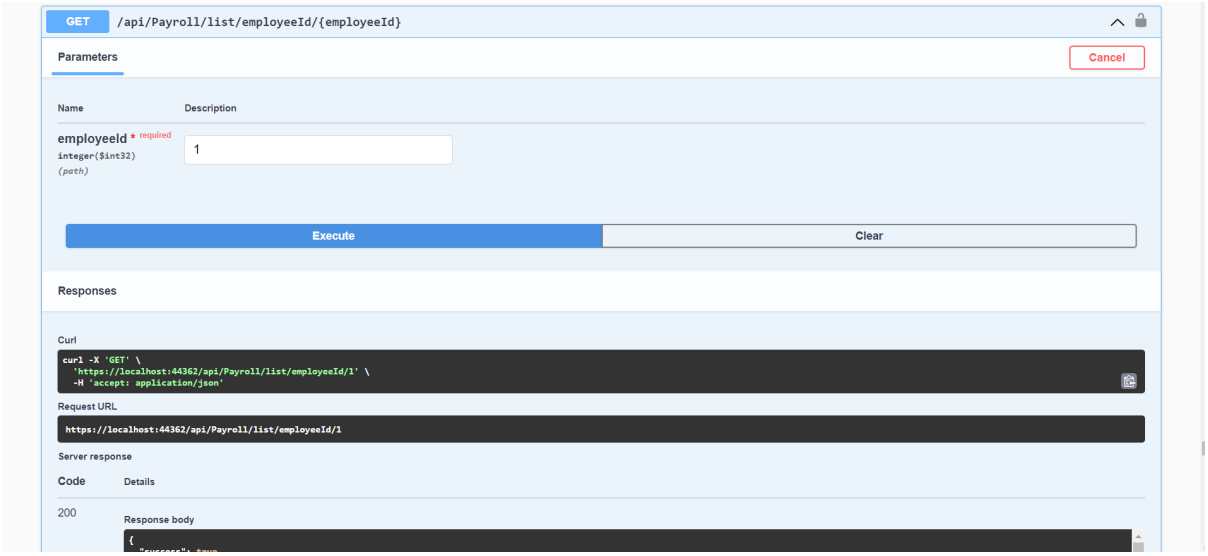
Payroll:

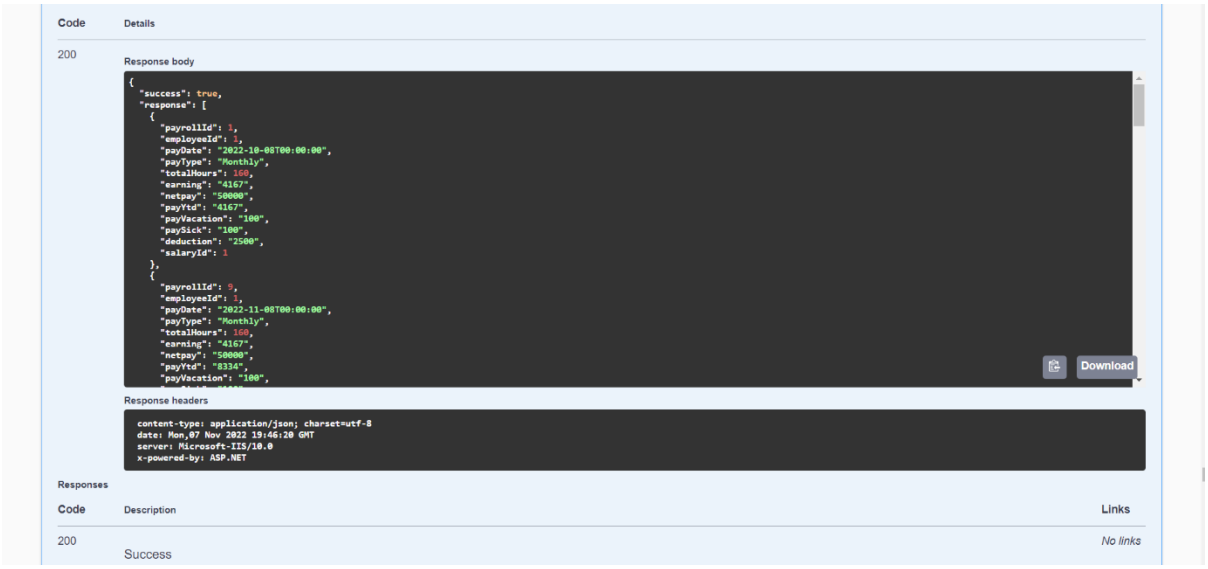
Payroll API gets the data of monthly salary associated to each employee. And holds the record of all the increments, deduction, Sick pay, Vacation pay, Number of hours an employee has worked on monthly bases. Calculating all these attributes gives the actual payment associated to each employee.



Payroll List with respect to Employee ID :

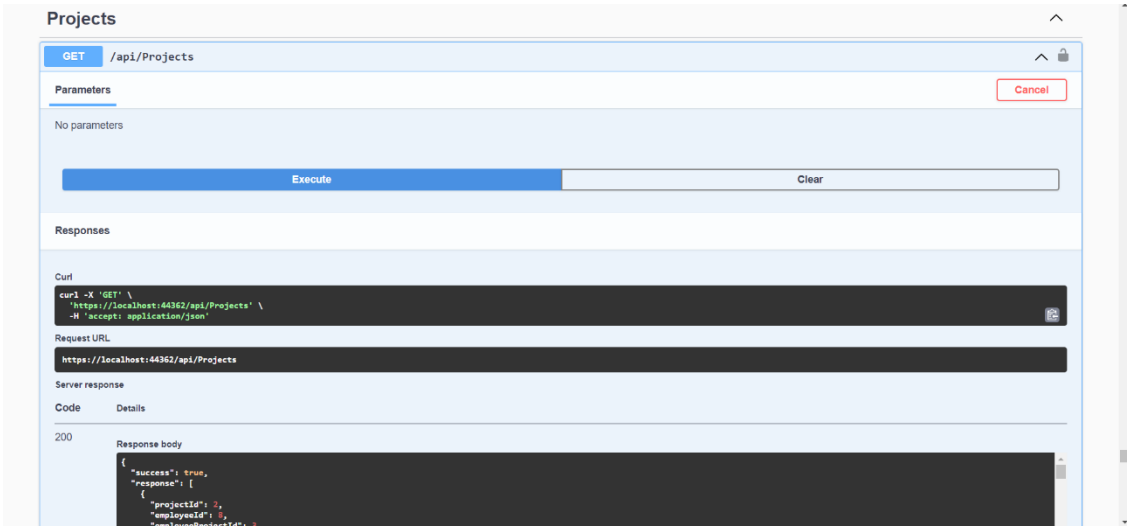
This API will use employee id and show all the information of payroll receipts that employee has received throughout the Year.

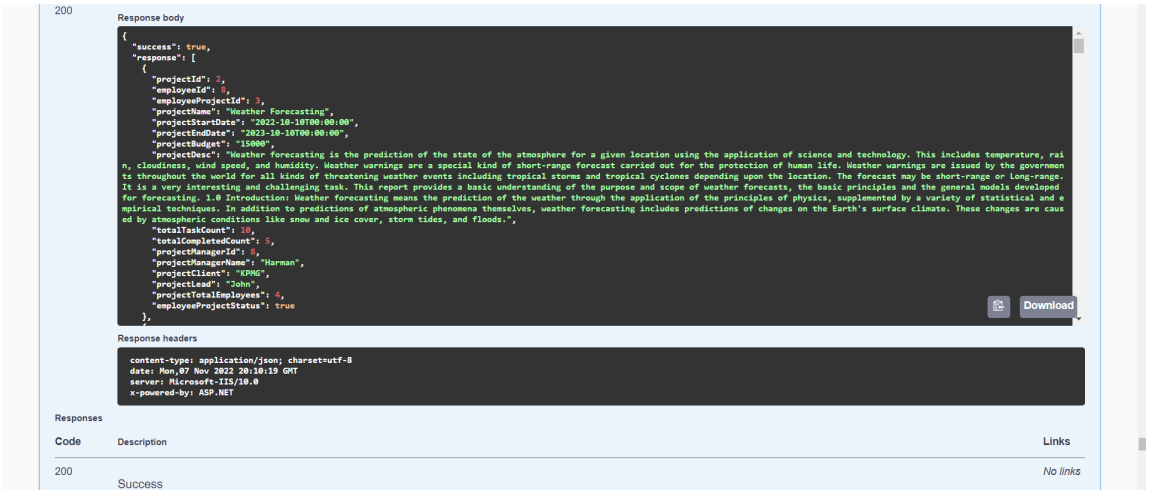




Projects:

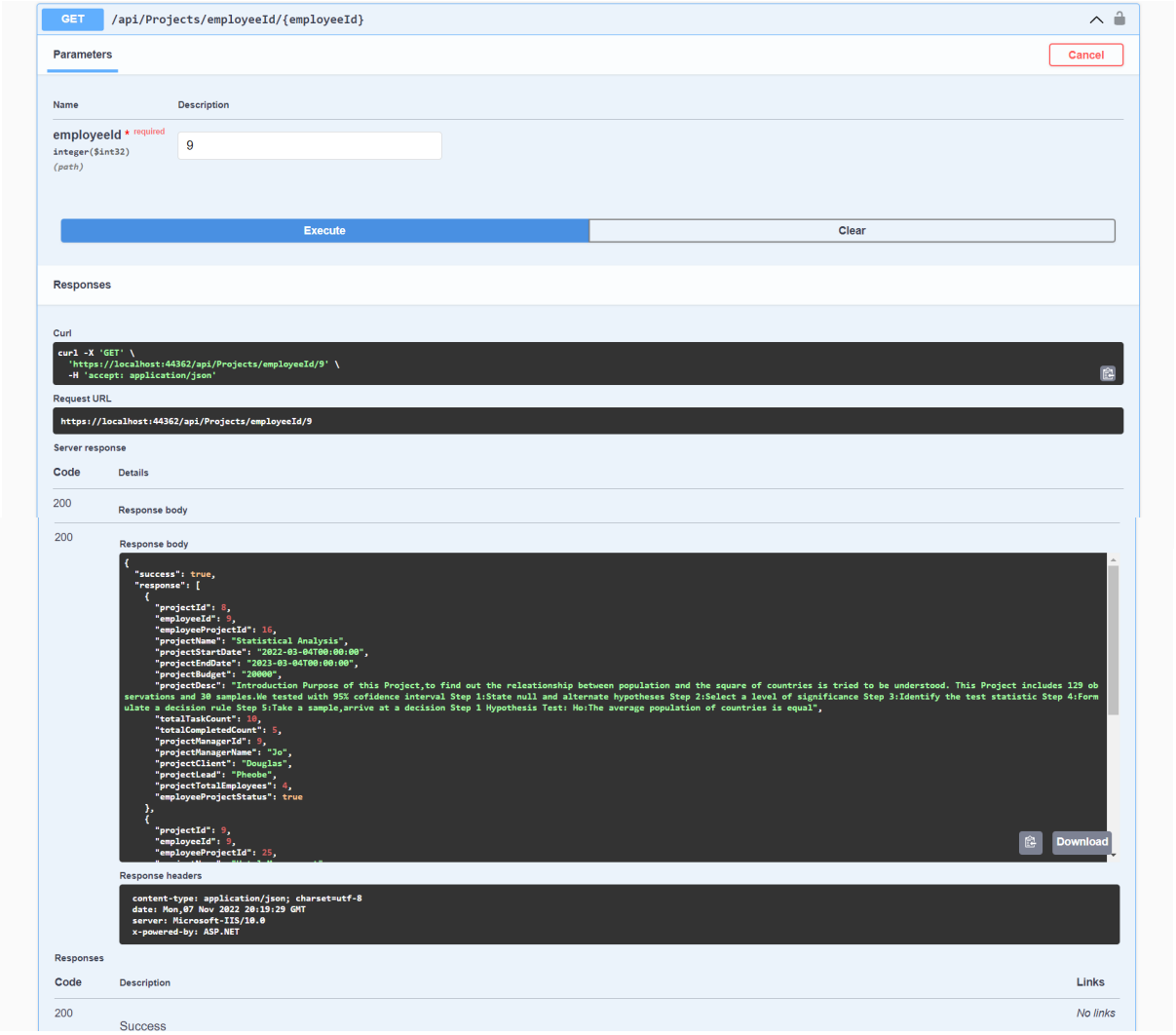
This will display all the projects that are undergoing in the company with detailed information associated to Employee Id and Project Id.





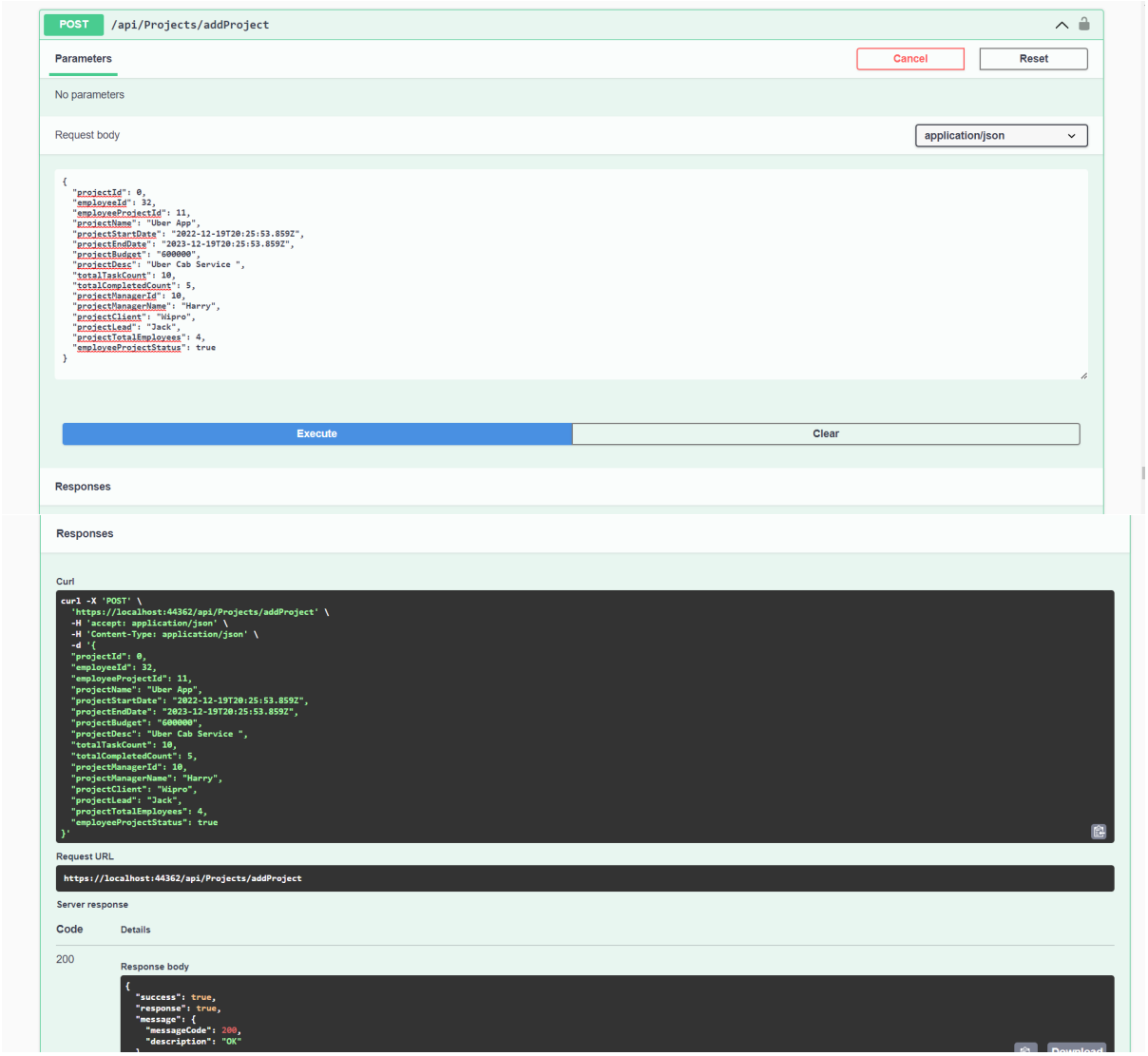
Project with respect to Employee ID:

This API will use employee id and display the projects that are associated to that employee. Managers are assigned more than one projects so this will give the information of all the projects a manager is monitoring over.



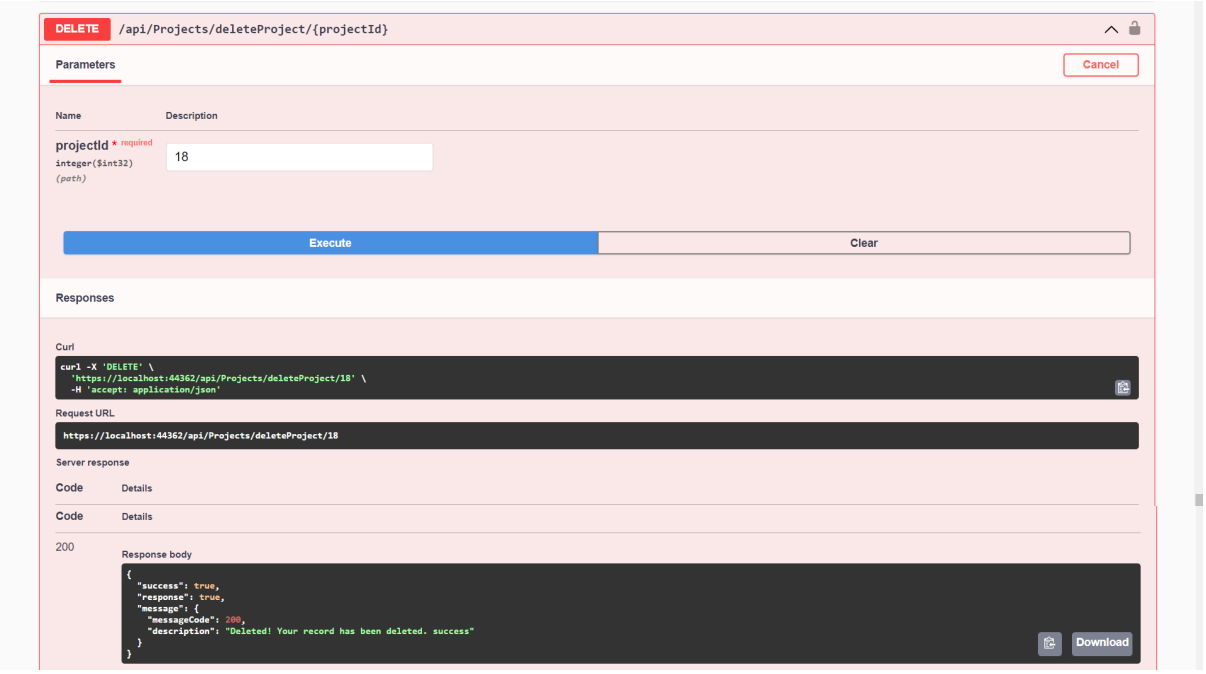
Adding a New Project:

This API will get information from Create Project screen and . Since, project Id is primary key this will be added automatically in the Database.



Deleting a Project:

It will ask user to enter the project Id that user wants to delete and it will erase the row associated to that project id in the Database.



Fetch API & Client side React code

After creating the front-end UI and back-end logic, from this stage we are focusing on connecting the front-end and back-end together. Although different React structures are still used widely, we chose the most popular and on trend one – hook, to manage the data flow. Since we have 21 tables and populating data takes time, the pages may seem empty. However, the data now is mostly dynamic and be able to maintain using CRUD.

Profile Update

- Fetching and putting employee data from/ to API
- Adding skills showing on page as badges
- After clicking on resignation, system will send notification to manager's email

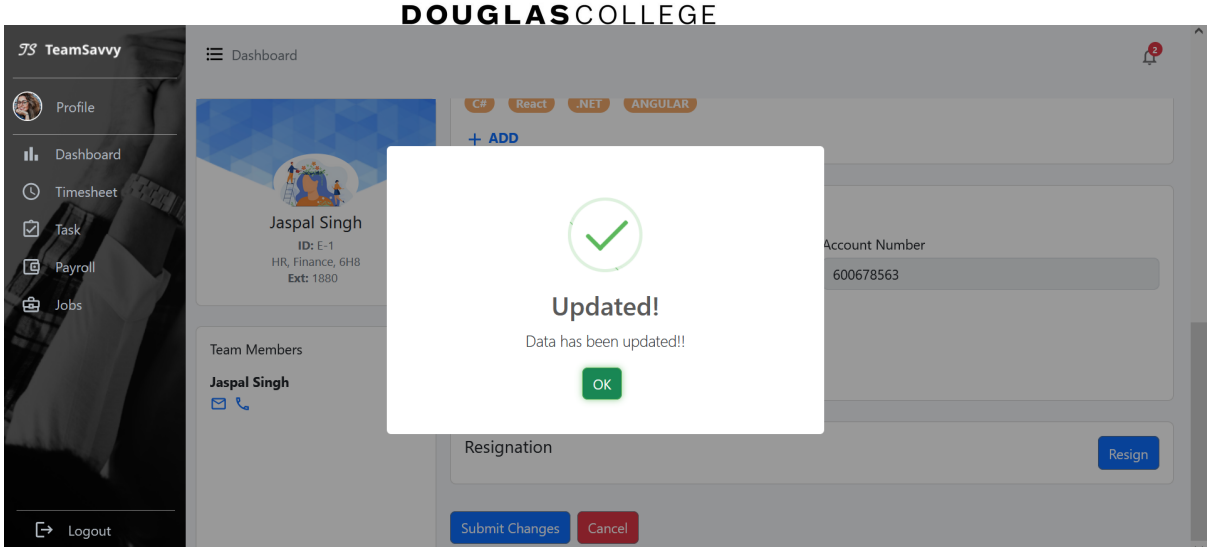


FIGURE 1: EMPLOYEE PROFILE UPDATE TO DATABASE

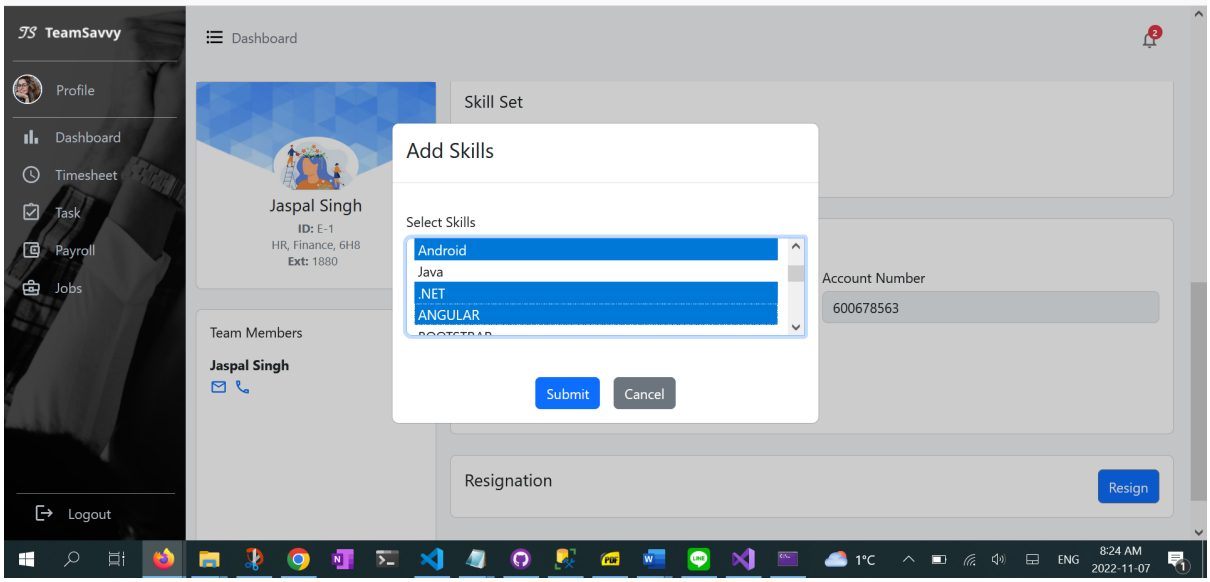


FIGURE 2: MULTIPLE SKILLS CAN BE ADDED AND DELETED

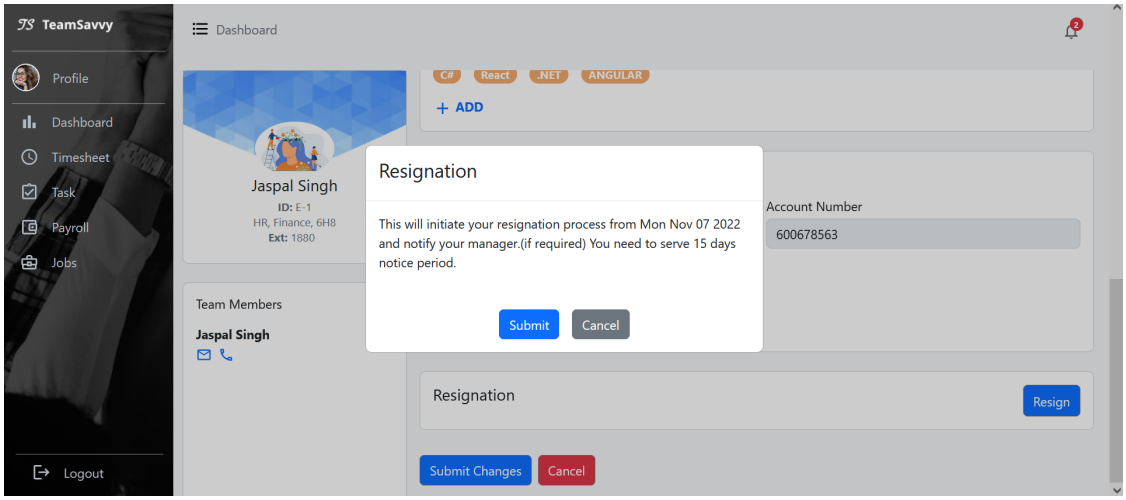


FIGURE 3:SIMPLIFY RESIGNATION PROCESS ALLOWING EMPLOYEE TO SEND AUTO RESIGNATION EMAIL TO MANAGER BY SIMPLE CLICK

Modify and validate task:

- Fetch data from api: get assigned jobs list
- Modal input validation: validate input at run time, preventing modal from closing if data is empty or not correct.
- Put request: update modified task to api using http put request

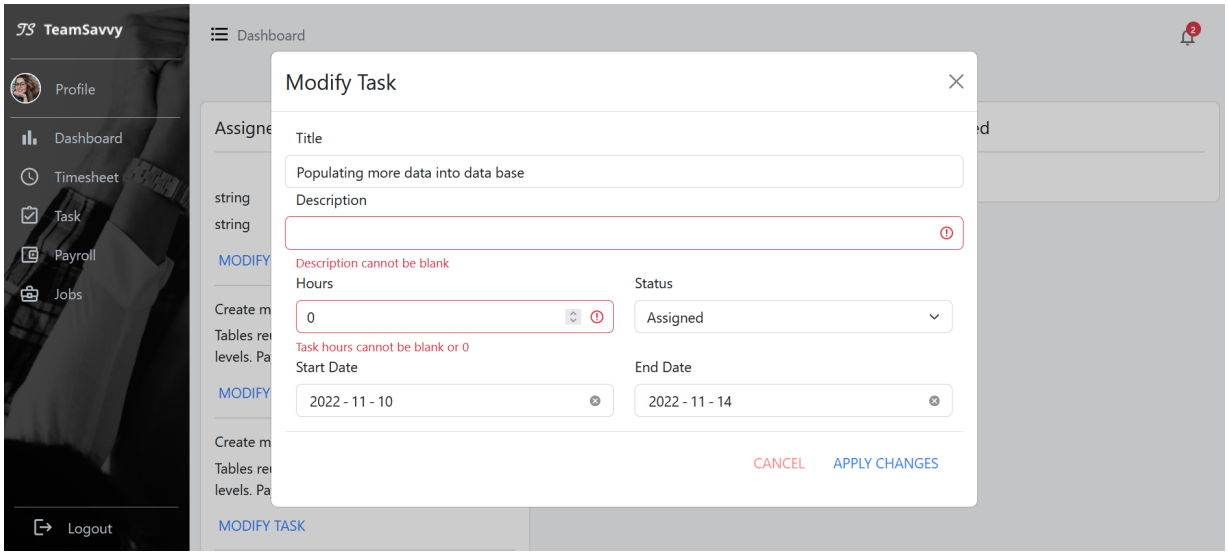


FIGURE 4: SIMPLE VALIDATION FOR INPUT. MODIFIED TASK WILL BE UPDATED TO DATABASE

Payslips and pay stub

Fetching data from payroll and employee API, using Links to pass prop and render detail in the child page

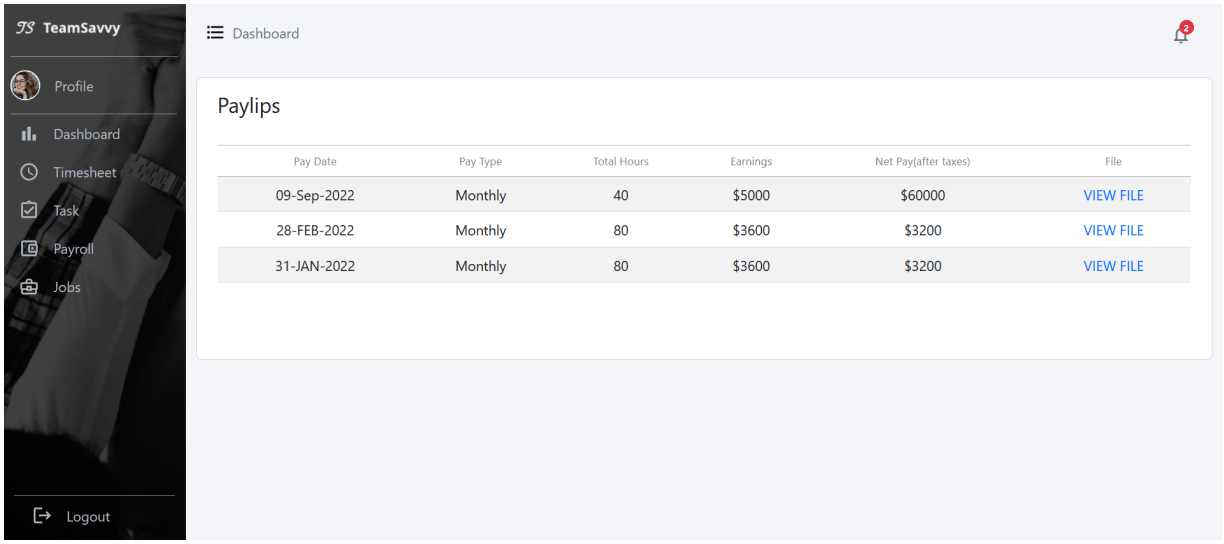


FIGURE 5: FETCH PAYSLIPS DATA FOR INDIVIDUALS. PAYSTUB DETAILS WILL BE SHOWN IN VIEW FILE.

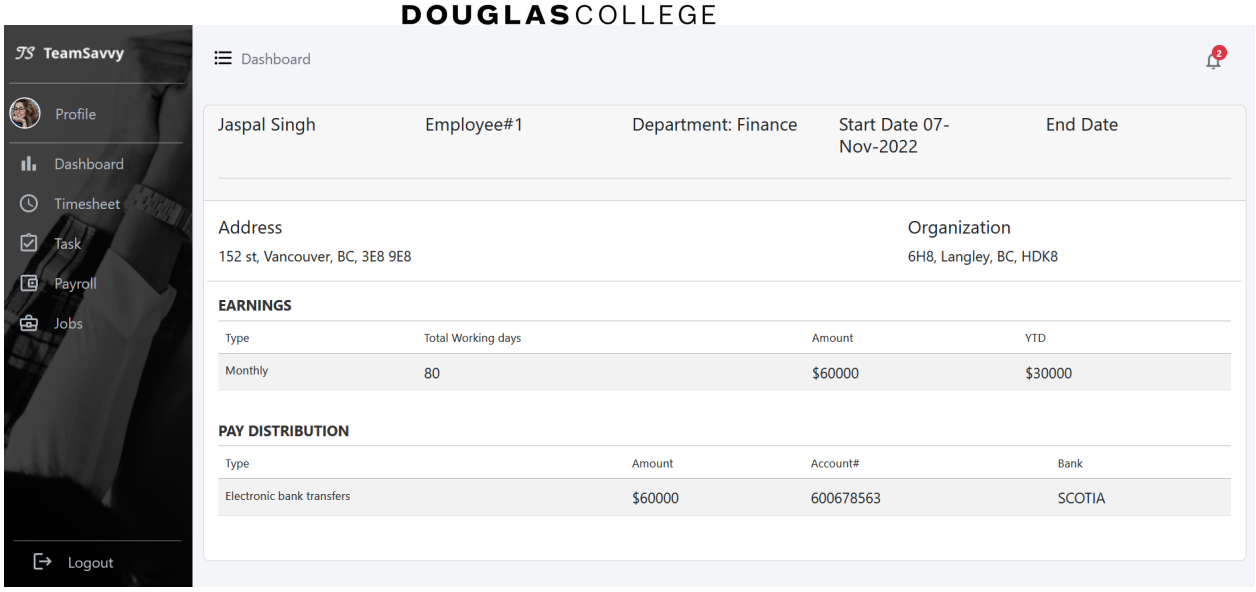


FIGURE 6: PASSING DATA FROM PAYROLL AND EMPLOYEE API

Jobs

- View change according to the employee role. For example, engineer won't be able to create jobs and see the applied details
- Fetching data from Jobs API, clicking on the view will show the corresponding detail on the right-side card.
- Clicking on Create Job shows the job creating card. The post request here hasn't implemented yet.
- Clicking on the applied button shows the employee applied for this job. The detail link will then show the employee's profile.

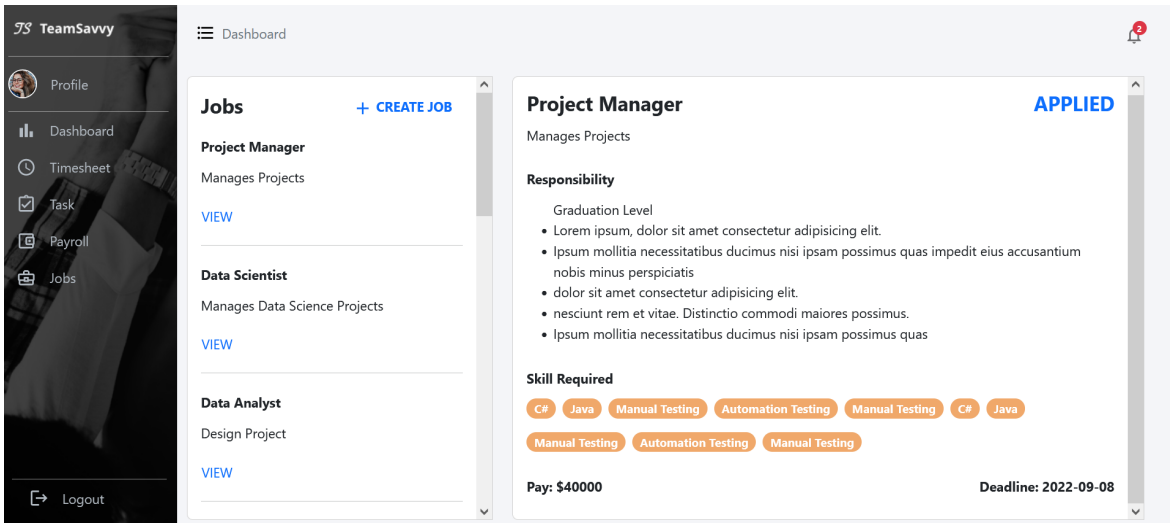


FIGURE 7: HR VIEW OF JOB PAGE

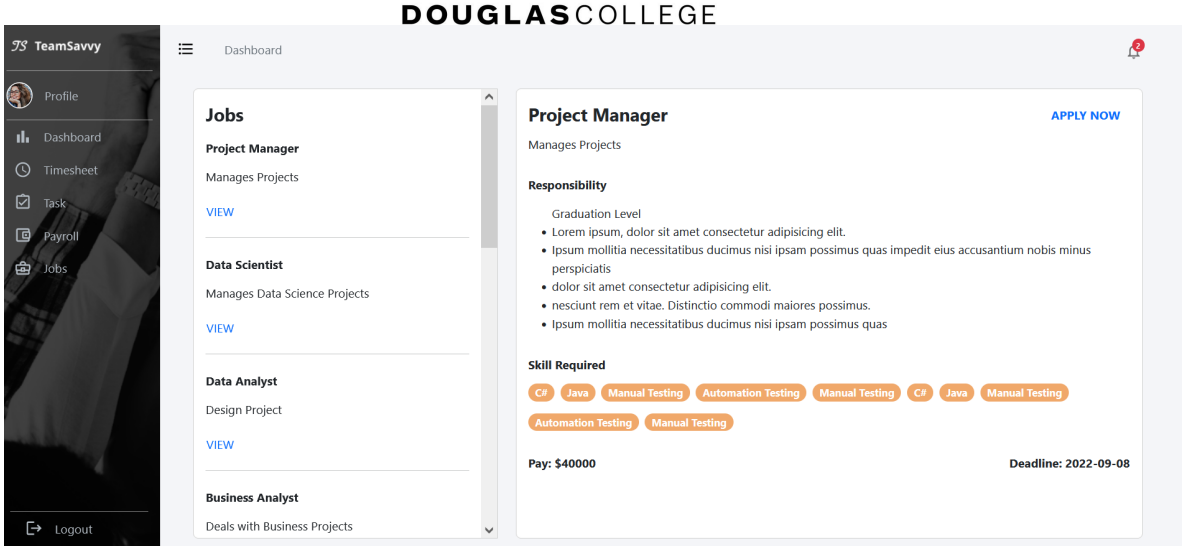


FIGURE 8:EMPLOYEE VIEW OF JOB PAGE

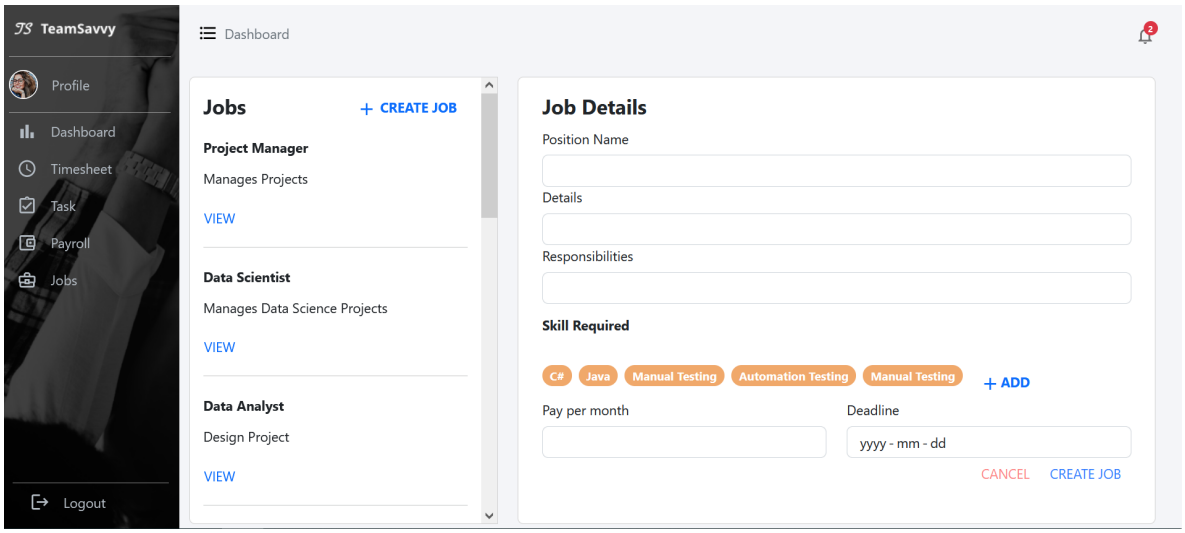


FIGURE 9: CREATE NEW INTERNAL JOB CARD

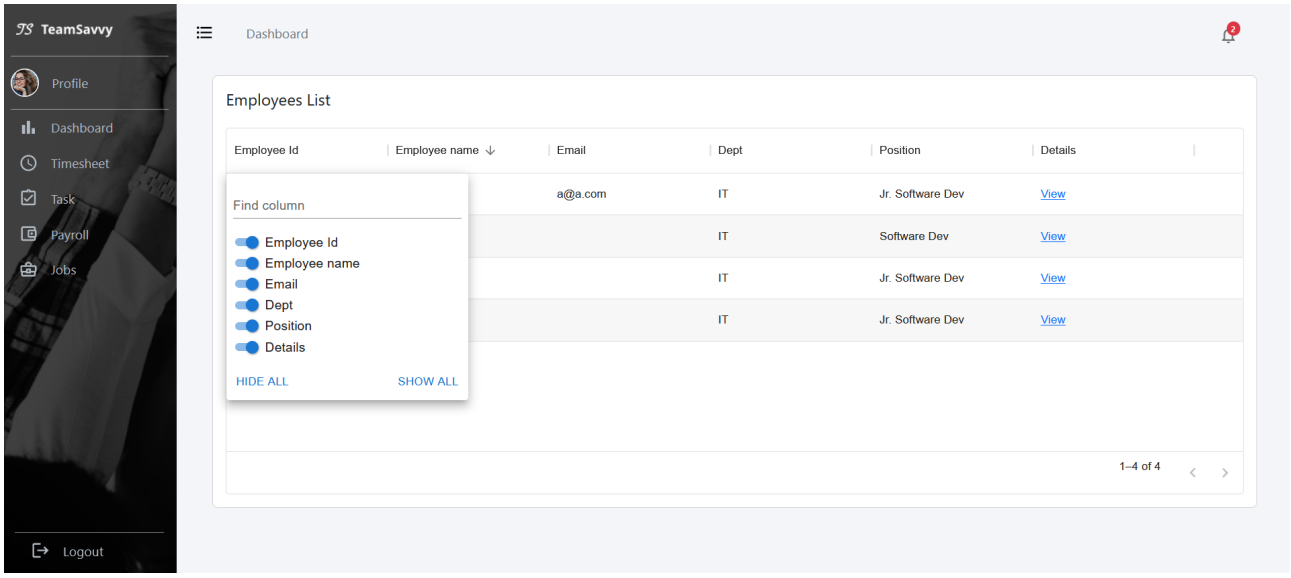


FIGURE 10: HR VIEW OF EMPLOYEES APPLIED FOR THIS JOB

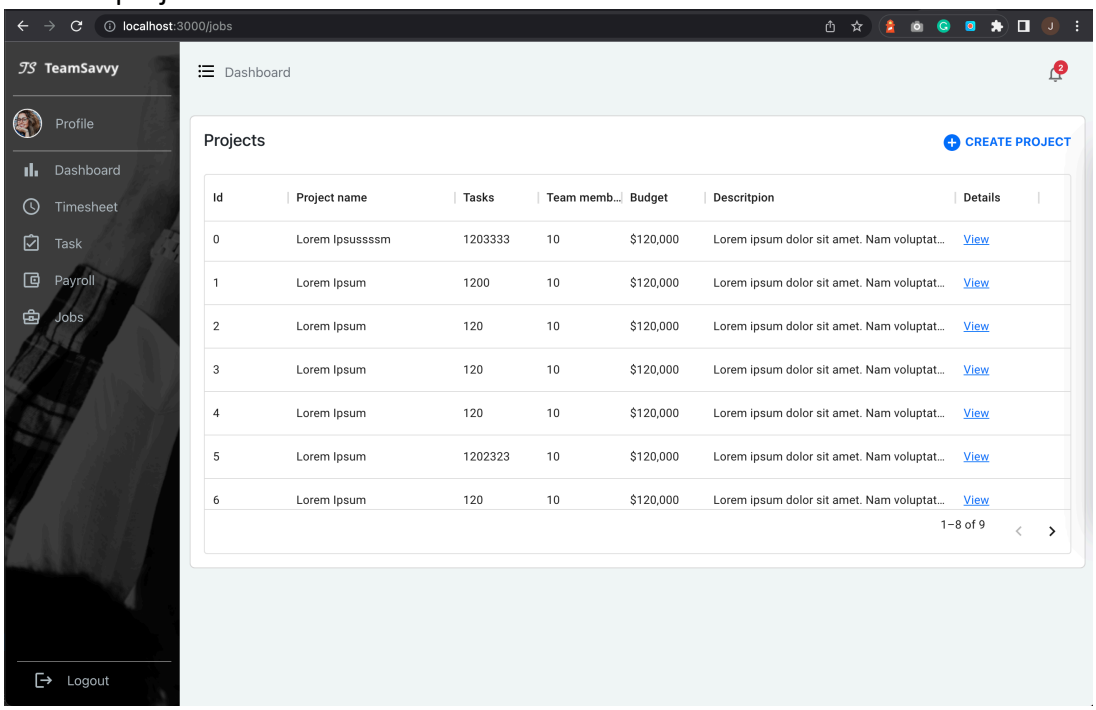
DOUGLAS COLLEGE

Front-end

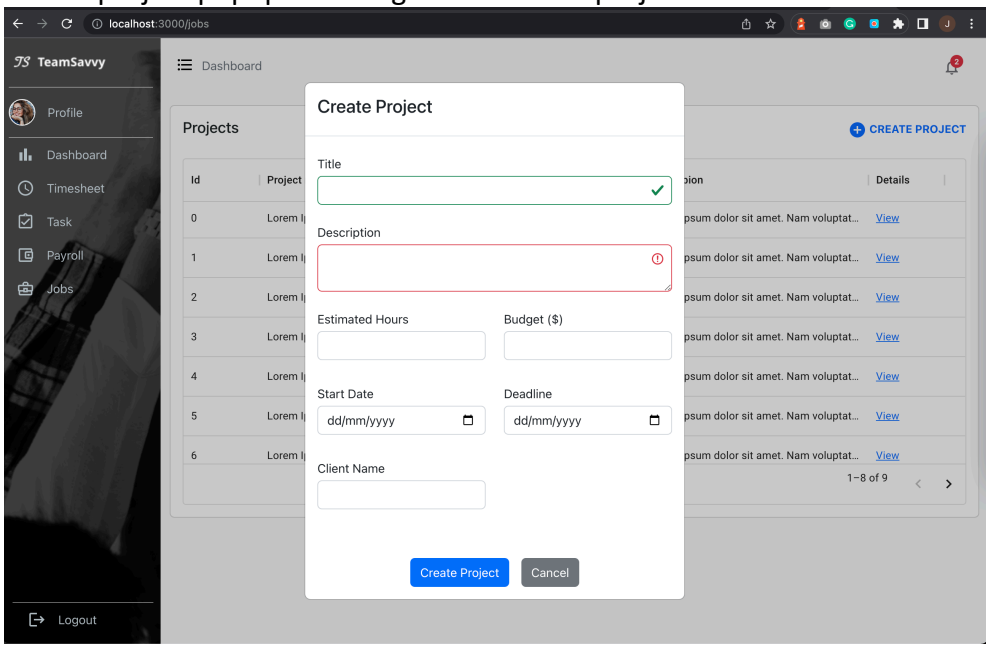
We started working with react js, bootstrap for developing our front-end with responsive screens, we decided to start with role 'employee'. Pages we have achieved so far:

Project List:

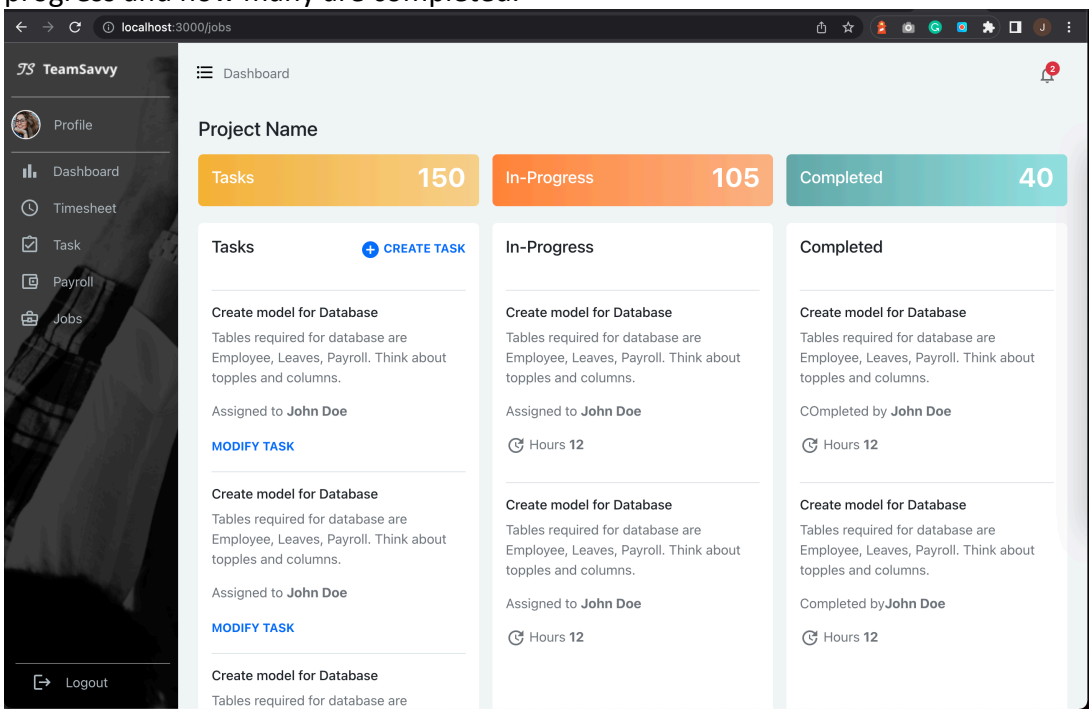
This list depicts the Information about the projects under manager or for HR total number of projects in organization. The screen will have project names, number of tasks assigned, number of team members in each project, project budget, project description and view link to show project details.



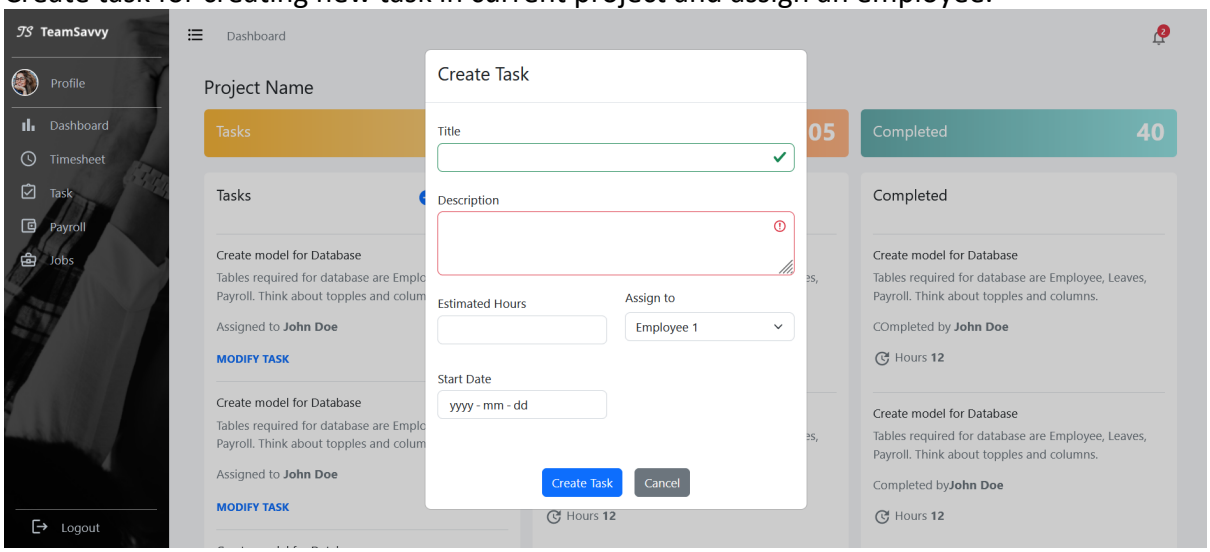
Create project popup for manger to add new project in the list



Project details page is for managers to assign tasks to employees in his team. It also has a simple statistic to manage the process of project by showing how many tasks are still in progress and how many are completed.

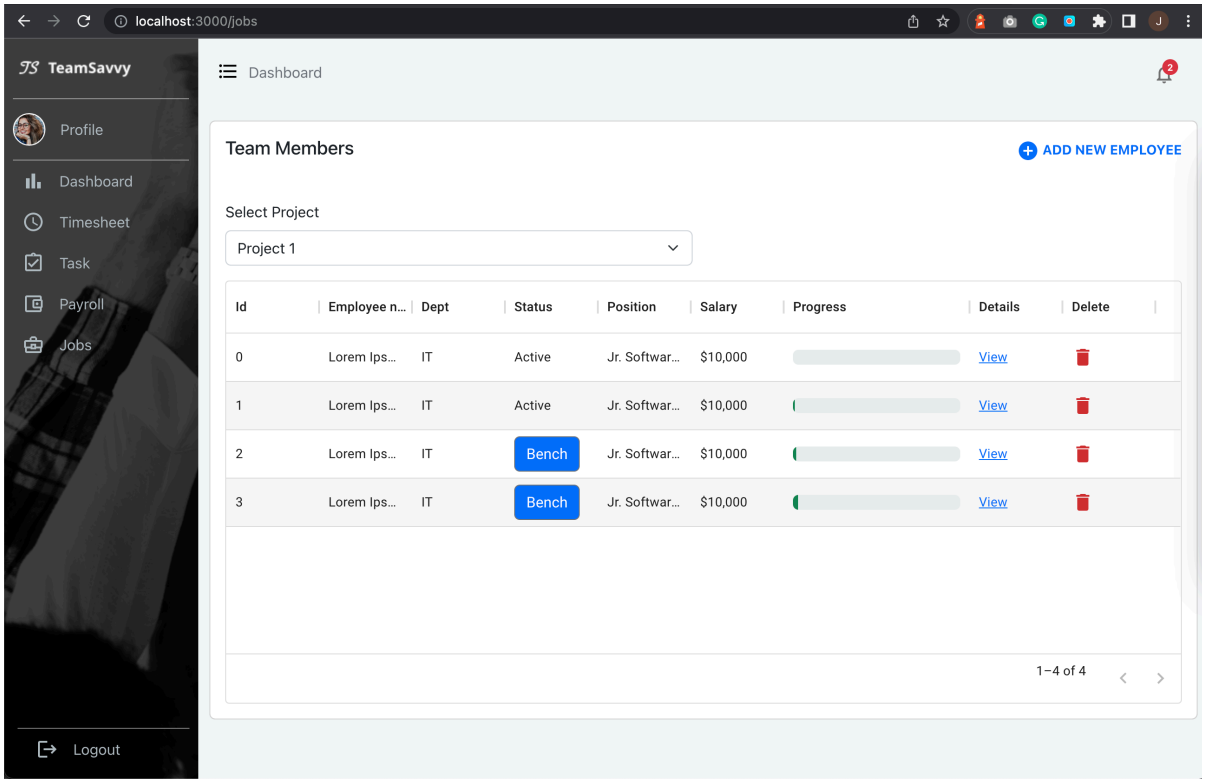


Create task for creating new task in current project and assign an employee.

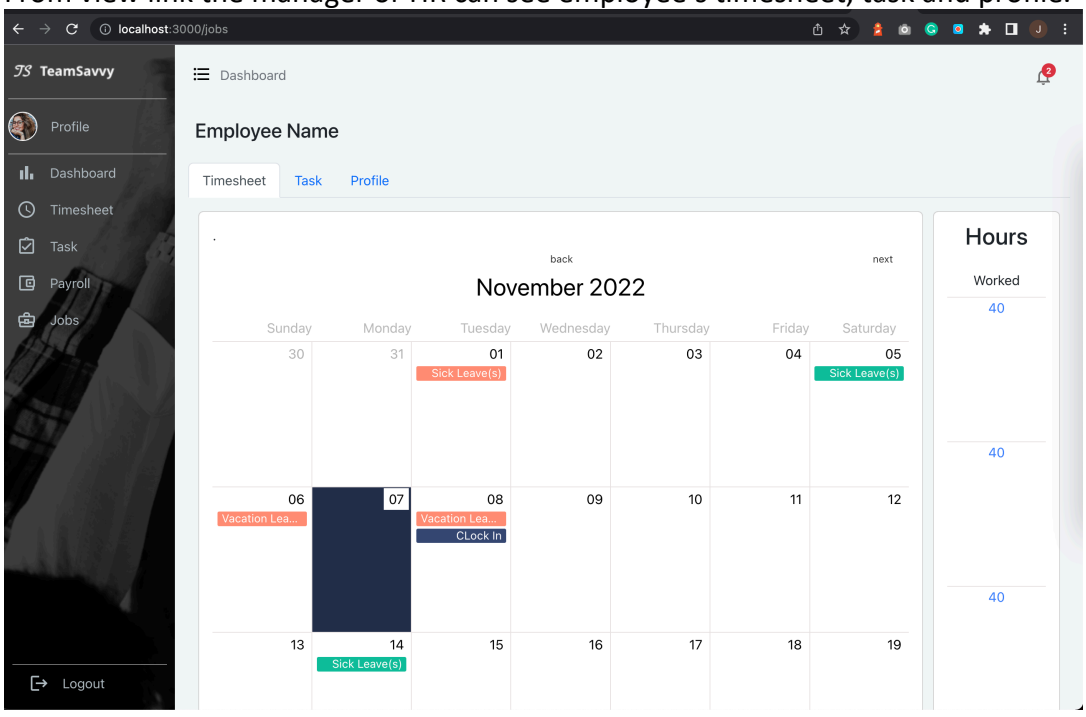


Team Members:

The screen will show the team members in the project selected from the list and Bench and Add new employee will only be shown to HR. It will have employee name, Dept , position, salary, progress and manager can delete the team member from this table and view button to show details about each team member.

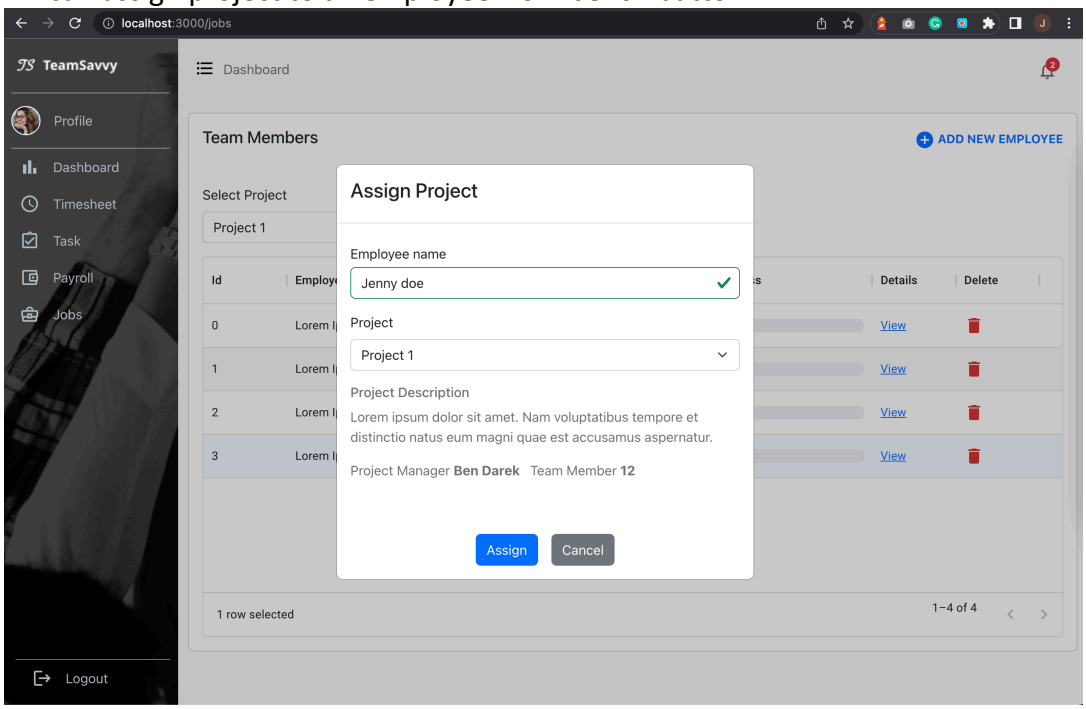


From view link the manager or HR can see employee's timesheet, task and profile.



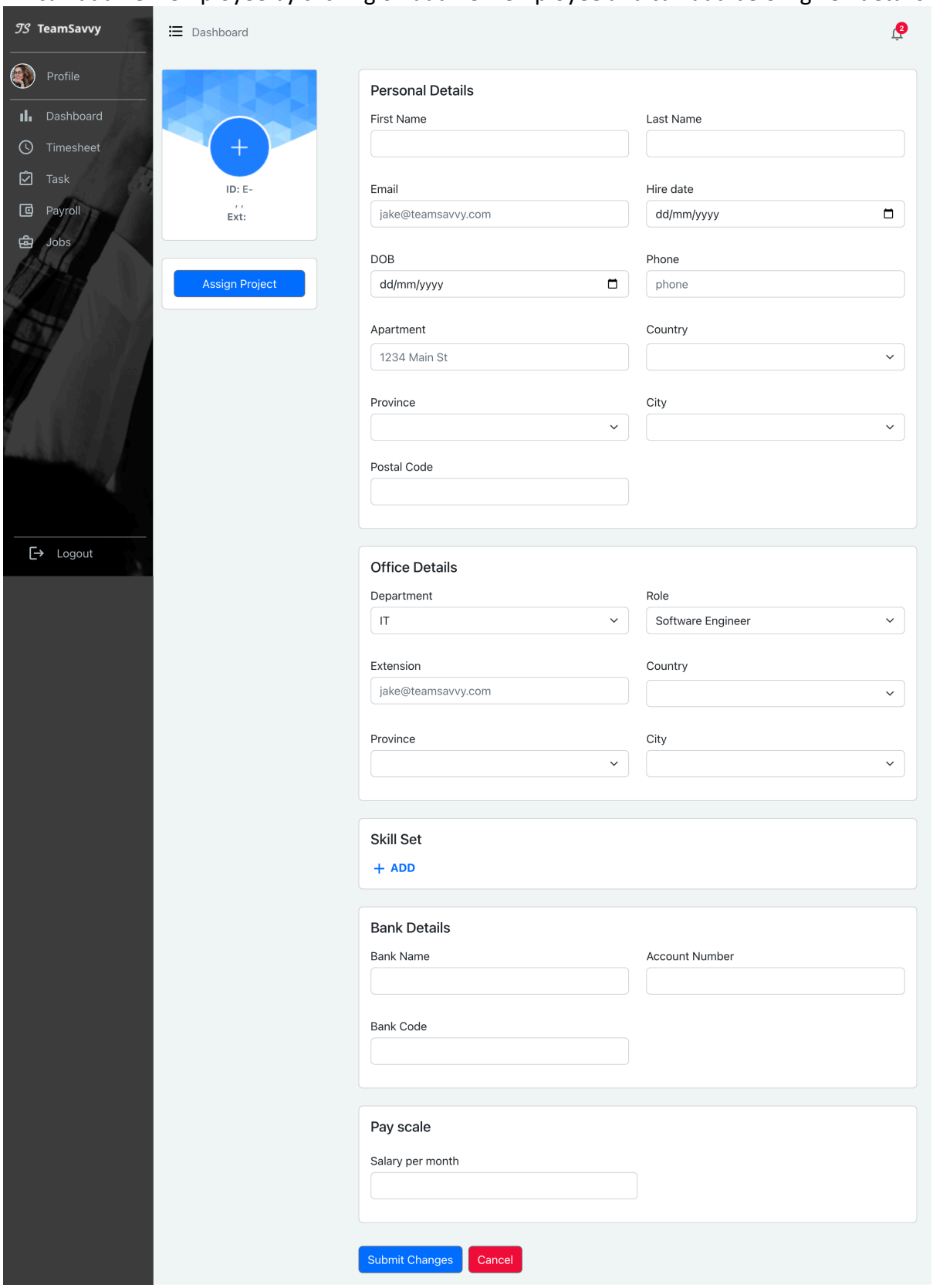
DOUGLAS COLLEGE

HR can assign project to an employee from bench button



DOUGLAS COLLEGE

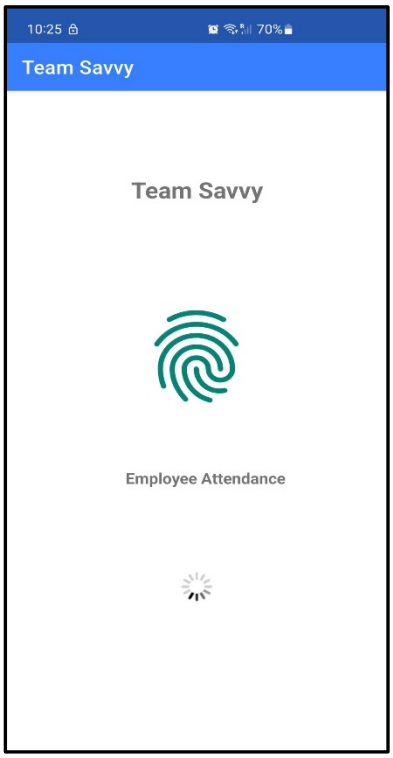
HR can add new employee by clicking on add new employee and can add below given details.



DOUGLAS COLLEGE

Previous Project Implementations:

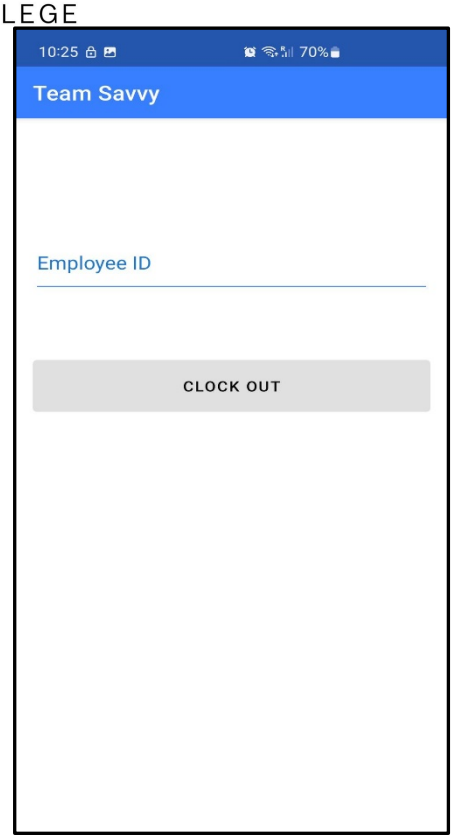
Android App for employee's biometric

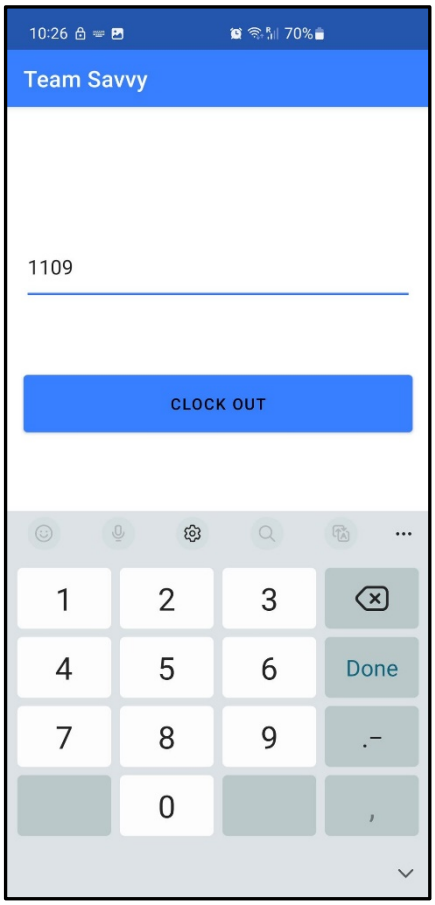


Landing screen for an employee showcasing the project name, image, message for an employee for the attendance and loader.

DOUGLAS COLLEGE

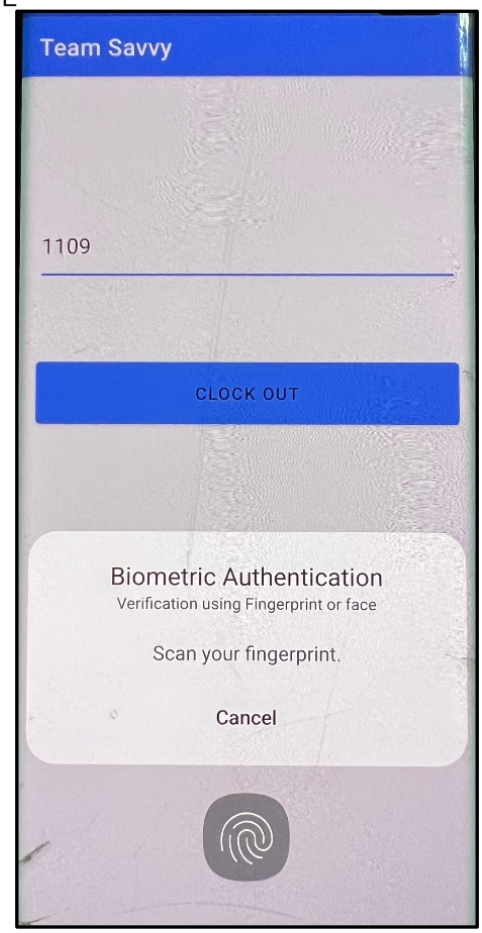
This screen will ask employee for their employee ID and allow him to clock-in or clock-out.
We are still working on design and few features.

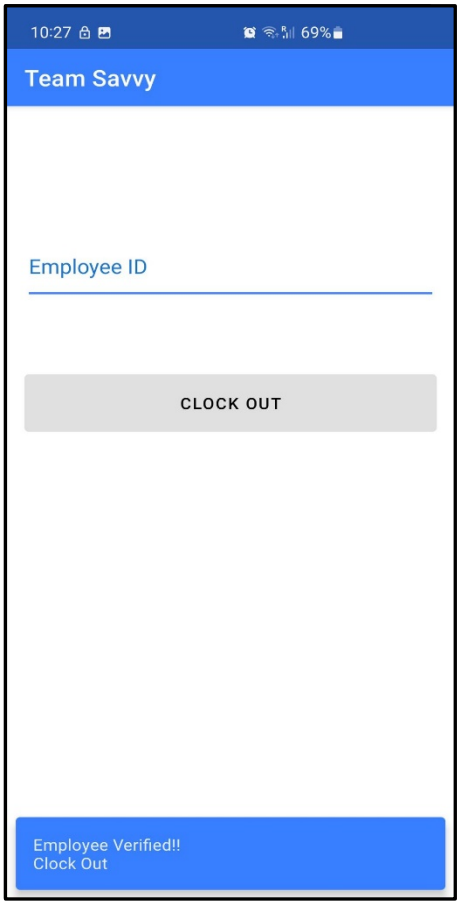




After entering the employee id the button will be activated.

The app will ask for biometric verification, if the fingerprint matches to the record of fingerprint in device the app will accept and allow user to clock-in or clock-out.



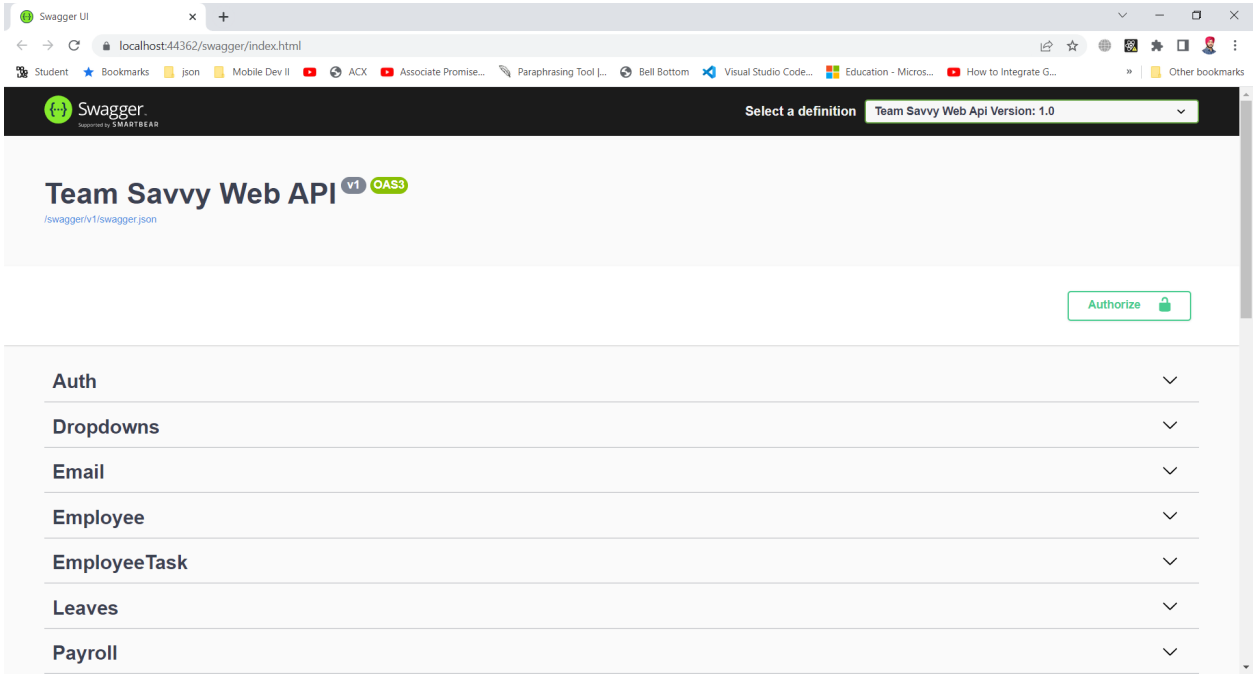


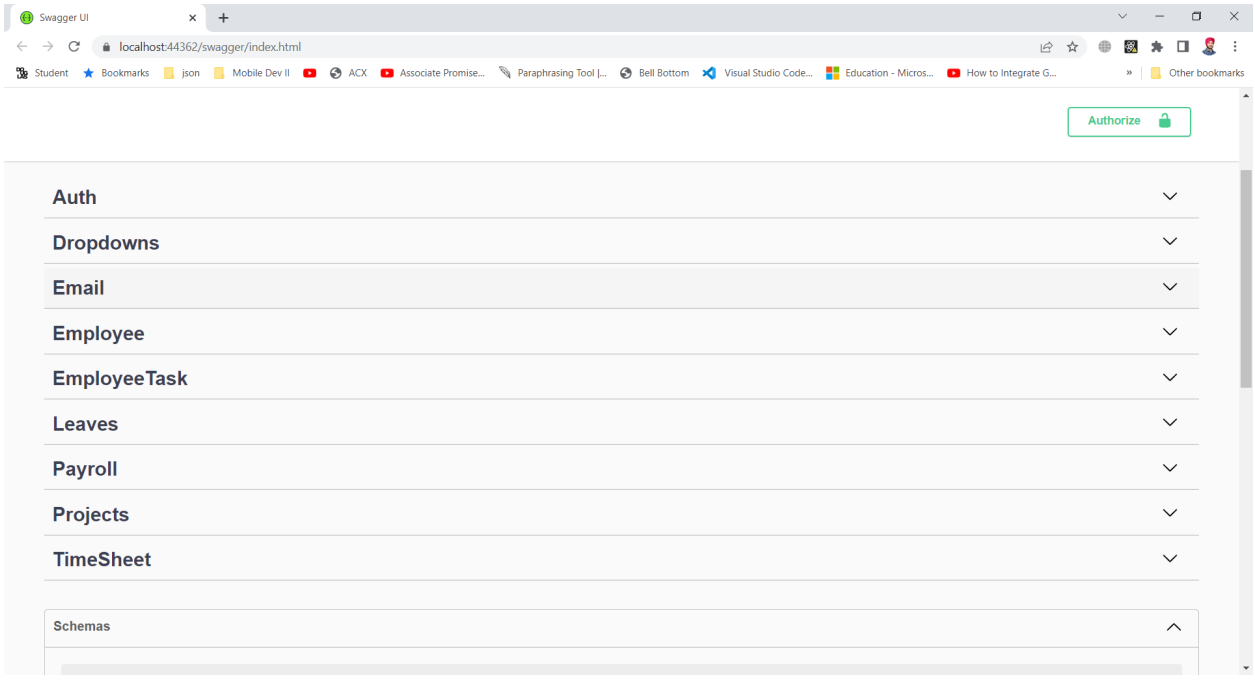
At last, the app will show the snack bar to show if an employee has been verified or not also displays the status clock-in or clock-out.

DOUGLAS COLLEGE

APIs

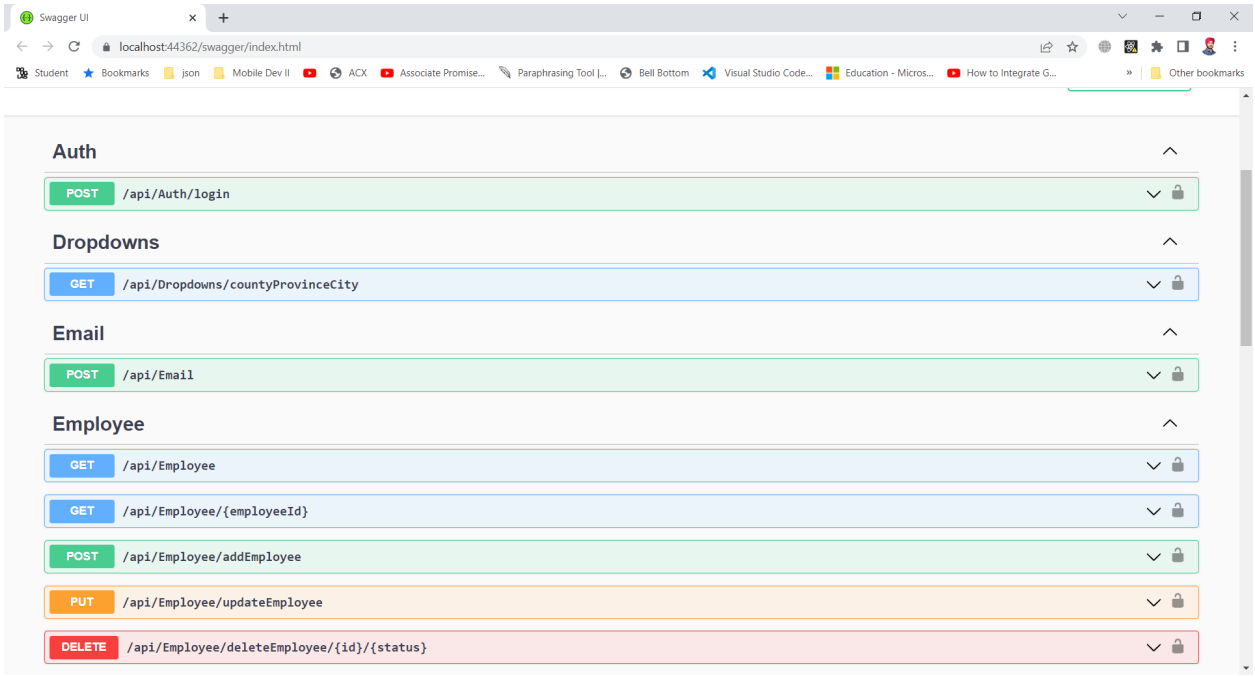
We have created API in .NET CORE using Entity Framework, LINQ, Automapper and swagger documentation.

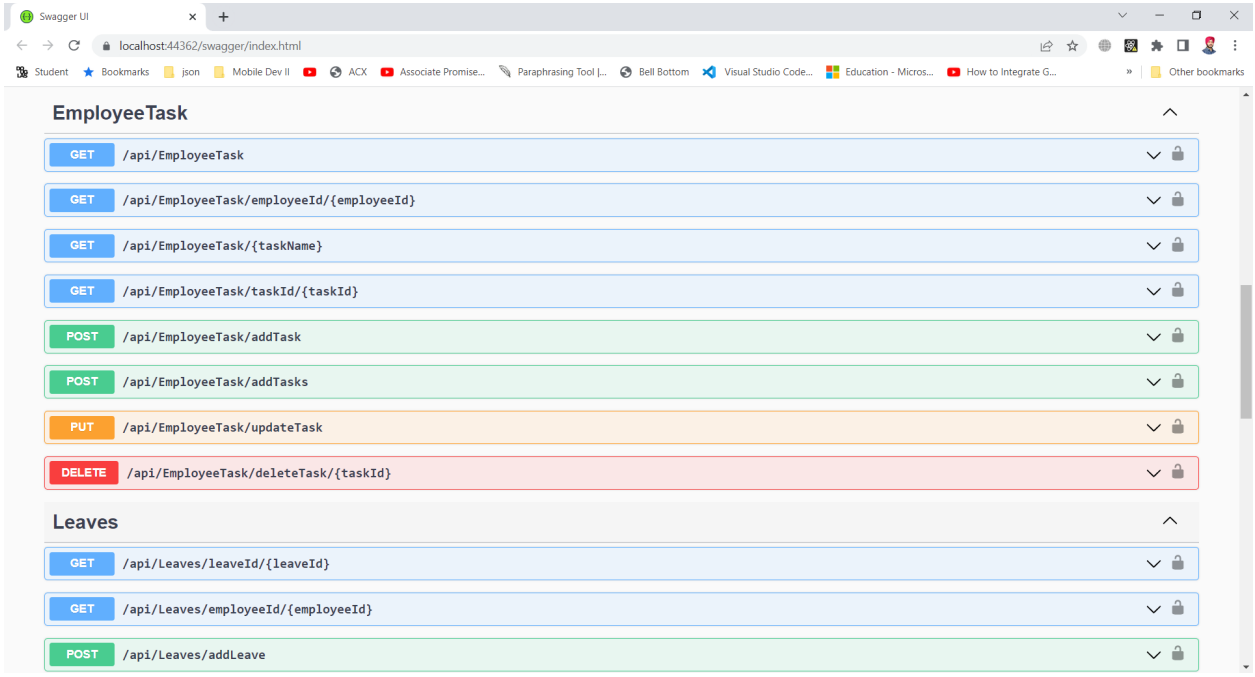


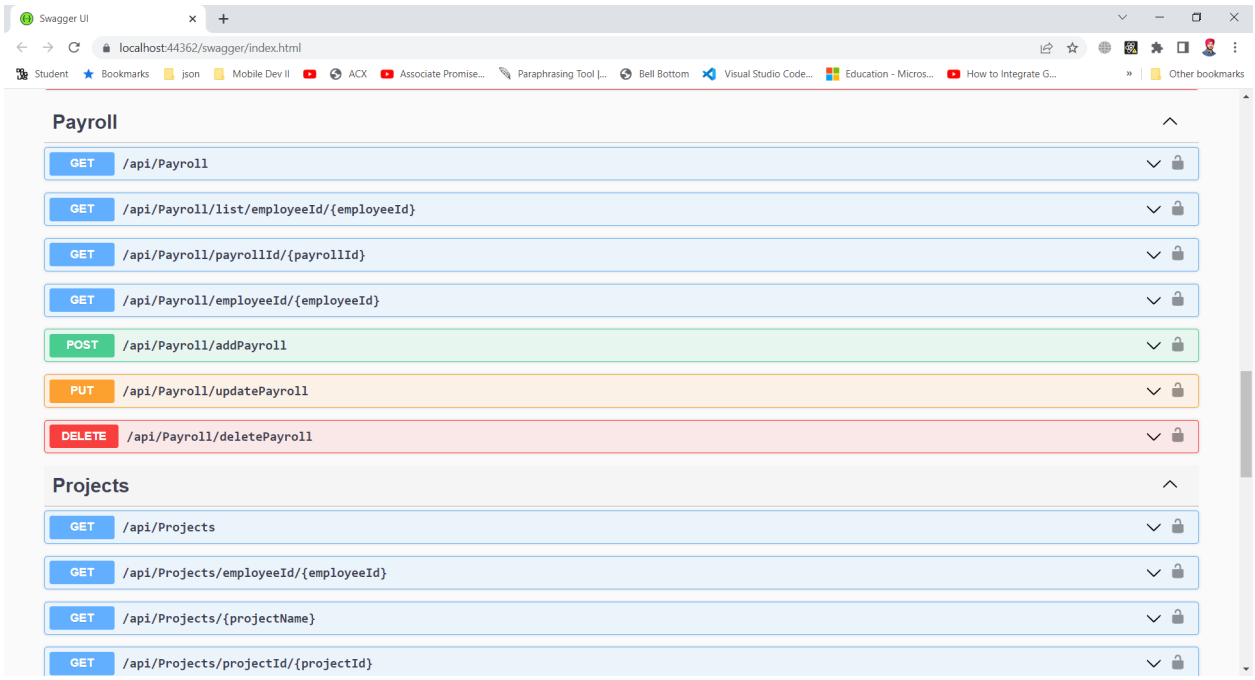


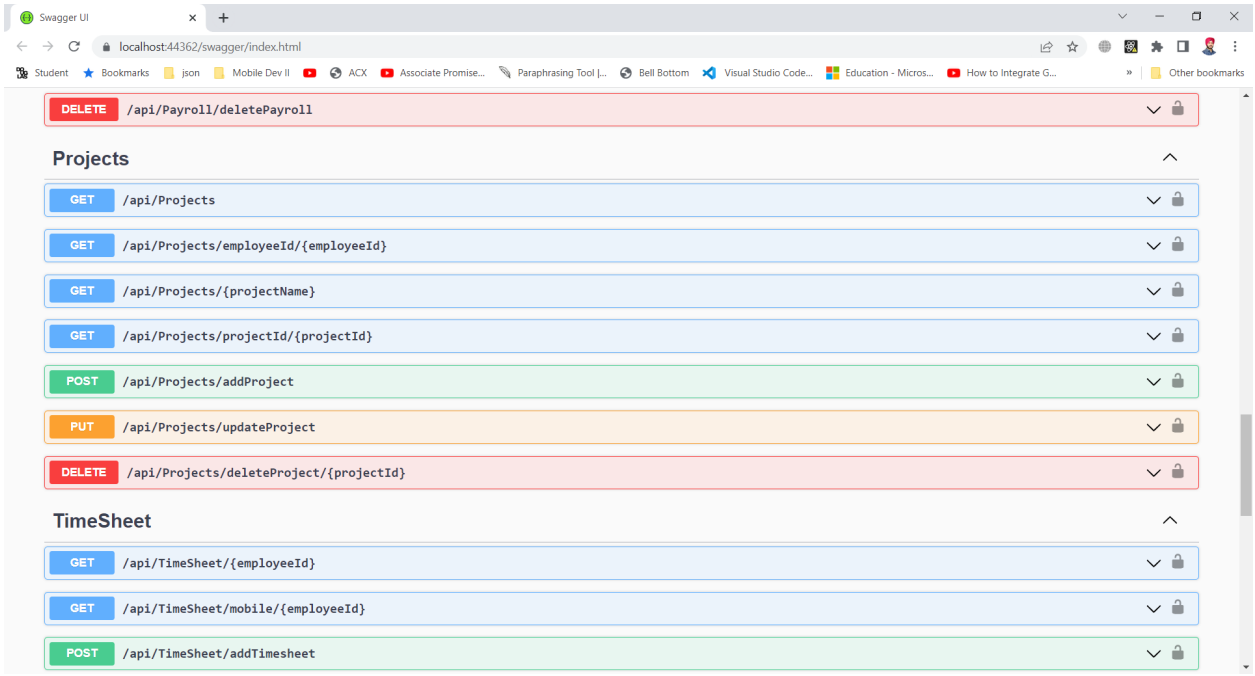
This is the swagger documentation to give a user interface to an API so developer can test API like postman. In this interface we can test all the request like post, delete, update, and get request. Above are the endpoints for Auth (Authorization), Dropdown, Email, Employee, Employee Task, Leaves, Payroll, Projects and Timesheet controllers.

DOUGLAS COLLEGE









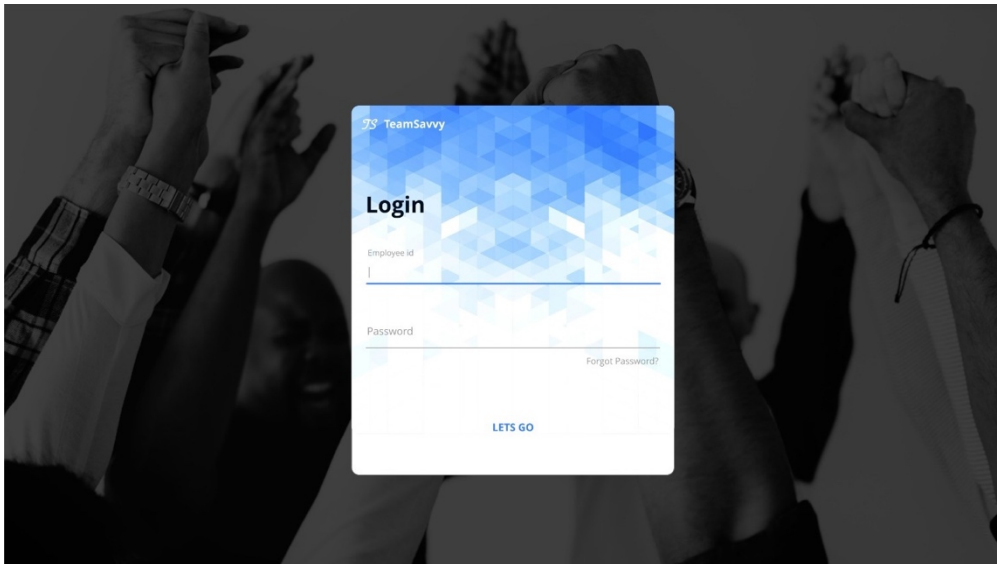
Above screenshots showing separate endpoints (GET,POST,PUT,DELETE) of all controllers created.

UI Design using Adobe XD

The snapshots below shows the flow of our application. Design Link: https://xd.adobe.com/view/732c0d35-1e41-41d7-876c-25bde1a844a9-267f/?fullscreen

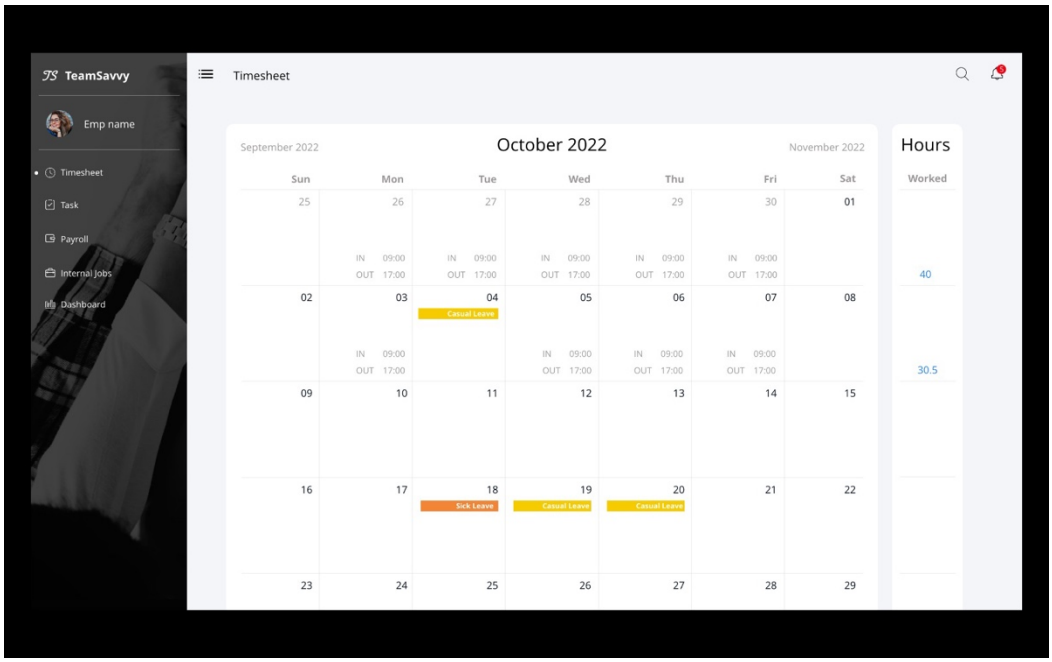
Login:

The employee and admin will login with the predefined credentials provided by organization.



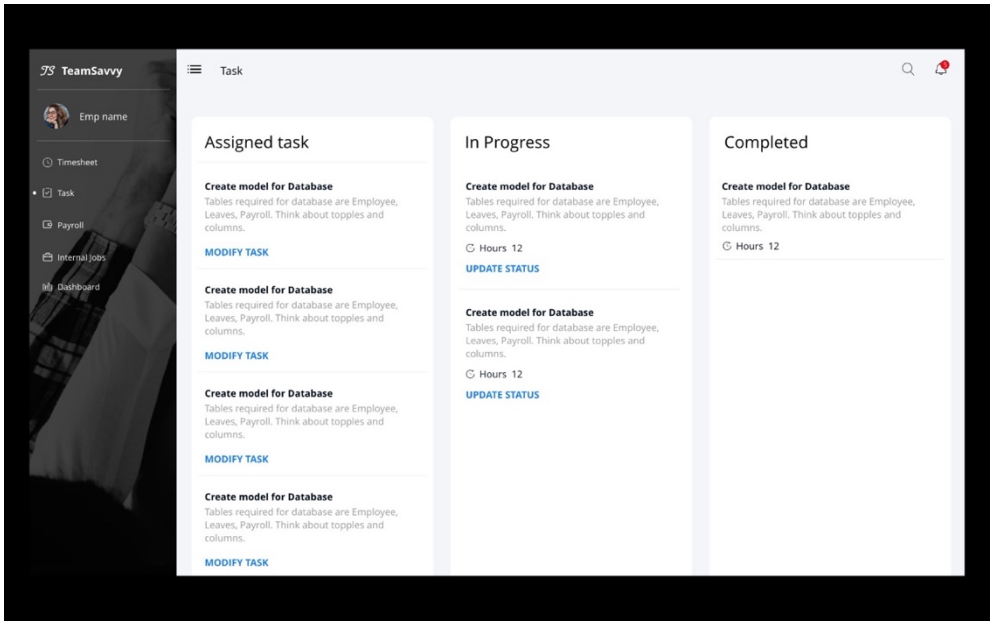
Timesheet:

This screen will show the information of employee clock-in and clock-out, with total hours they worked per week. Additionally, employee can apply for leaves and it will be displayed on the calendar



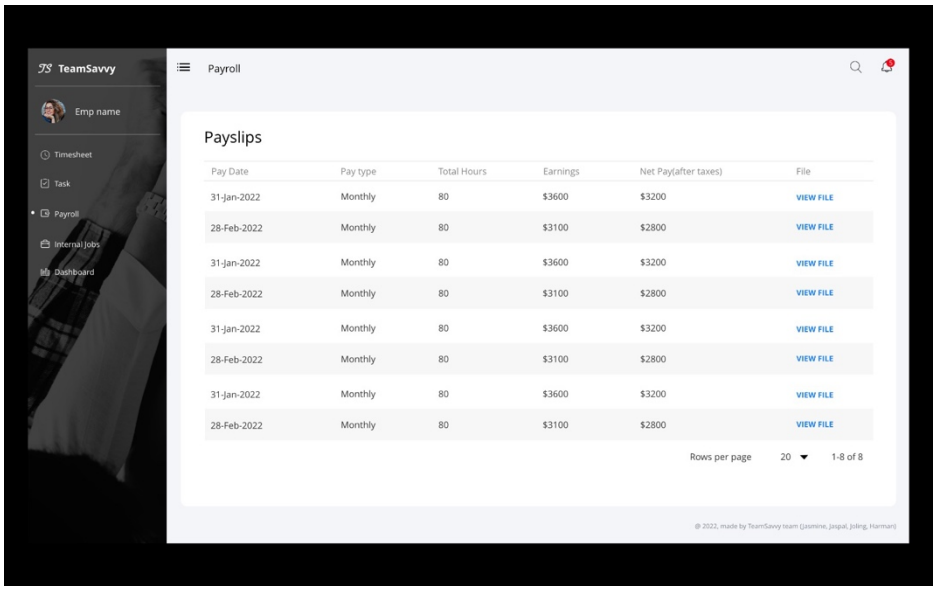
Task:

Task screen will have the assigned task for an employee, which he can modify to add hours, start date, and end date. Employee can also change the status from assigned to in-progress and then by using update status in in-progress lane he can change status to completed.



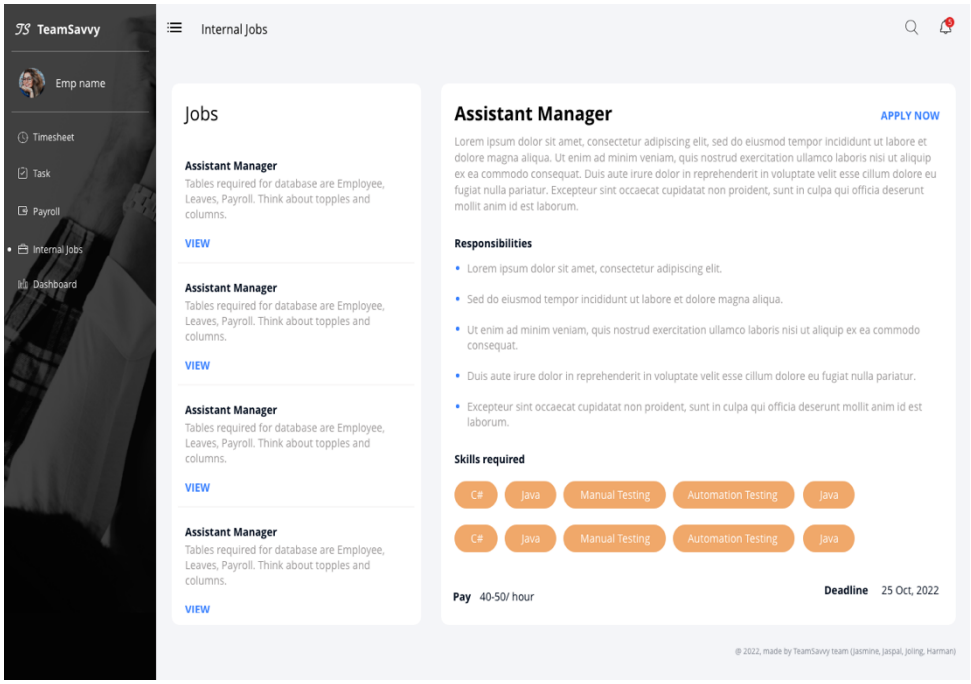
Payroll:

This screen will have the collection of pay slips with information of each month, total hours, and earnings, from here user can view each pay slip. The Pay Slip contains all the Earnings of an employee at the company from the date he has joined.



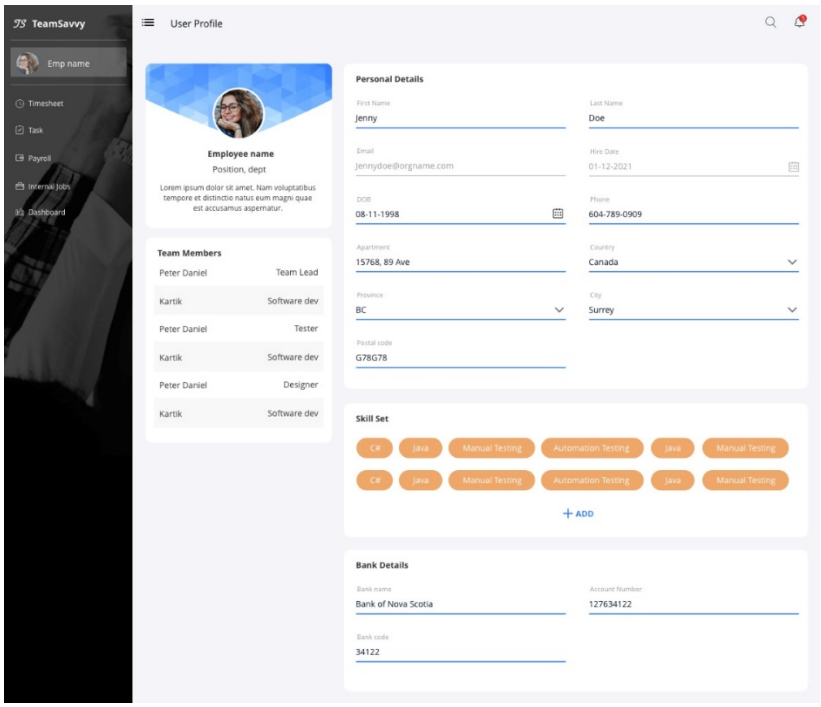
Internal Jobs:

This screen will display employee with the internal jobs postings which he can read the description and apply for the jobs.



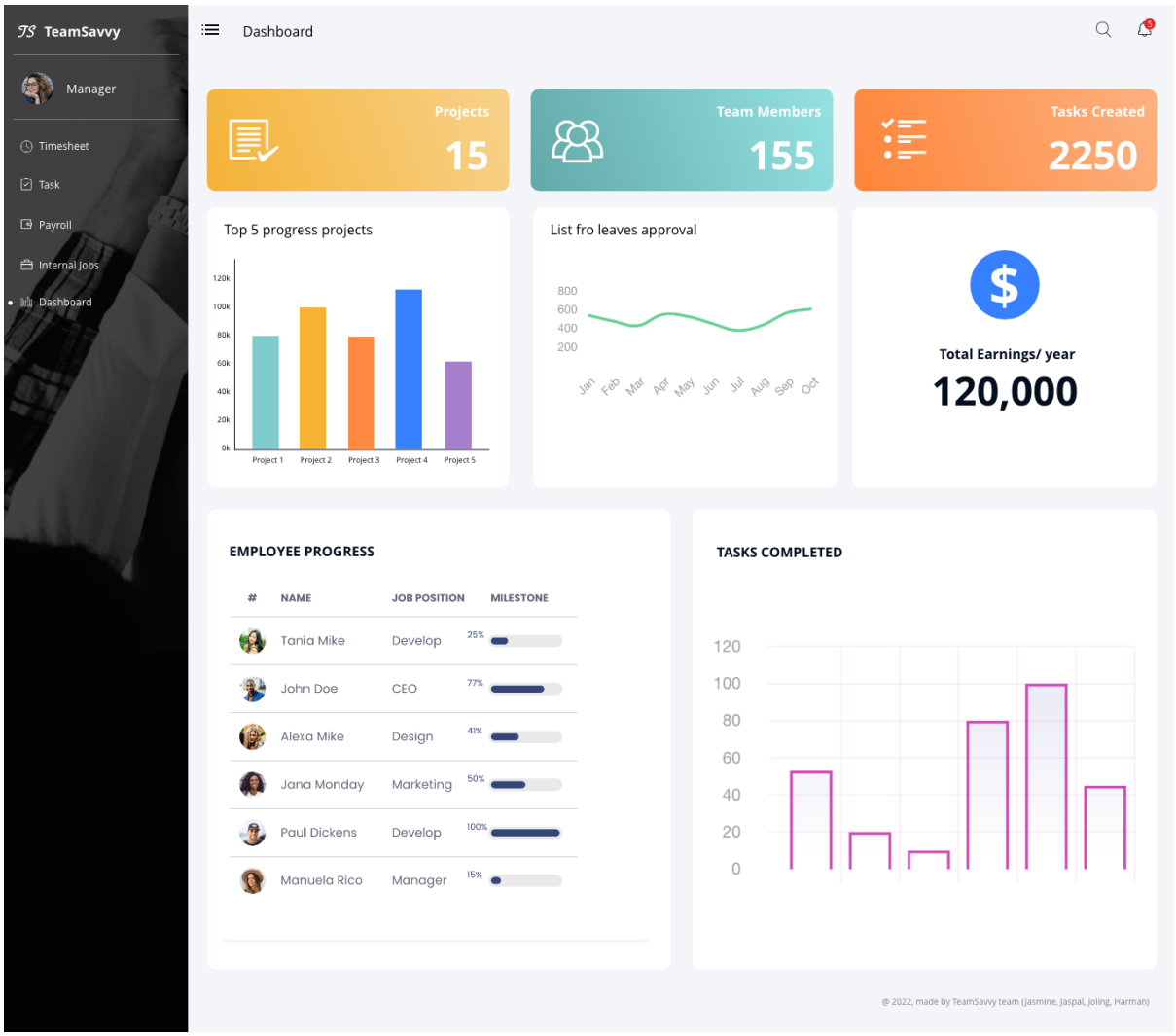
User Profile:

This screen will show the personal details, team members, position, department, skill set and bank details of an employee.



Dashboard:

This screen is for manager and HR, here they can view projects, team members, tasks, and other information. This screen will also have feature to create graphs widgets with predefined dropdowns. Manager can download, modify and delete the graph.



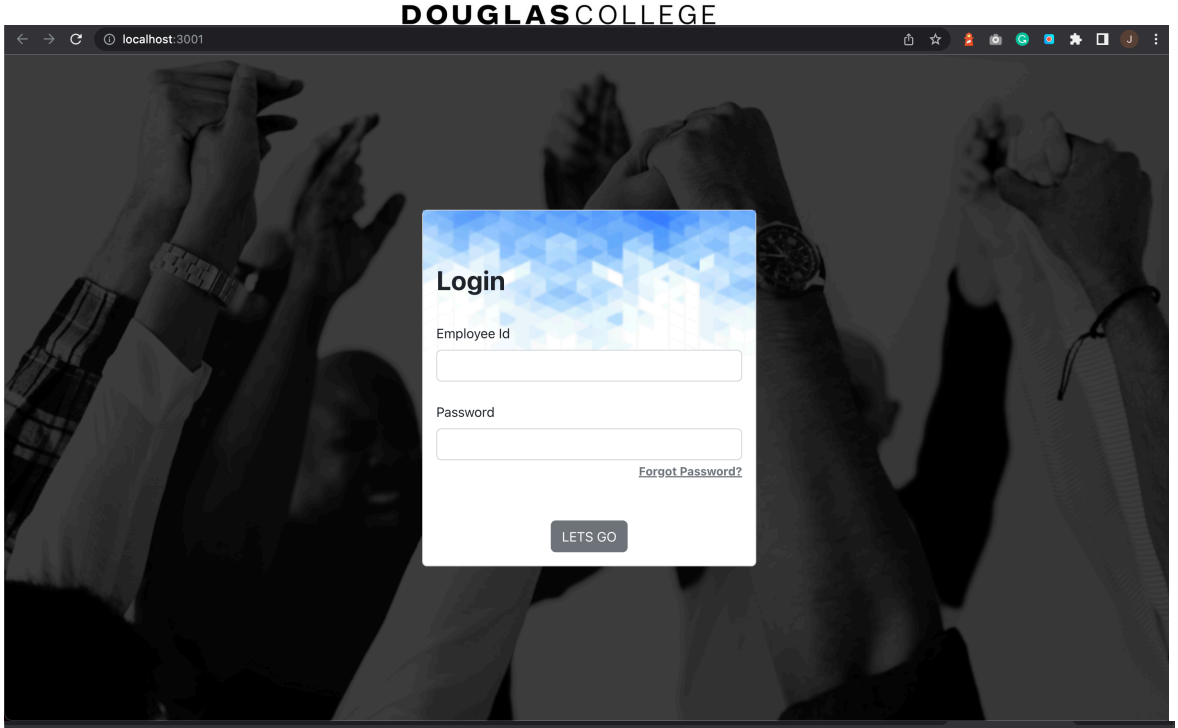
Front-end

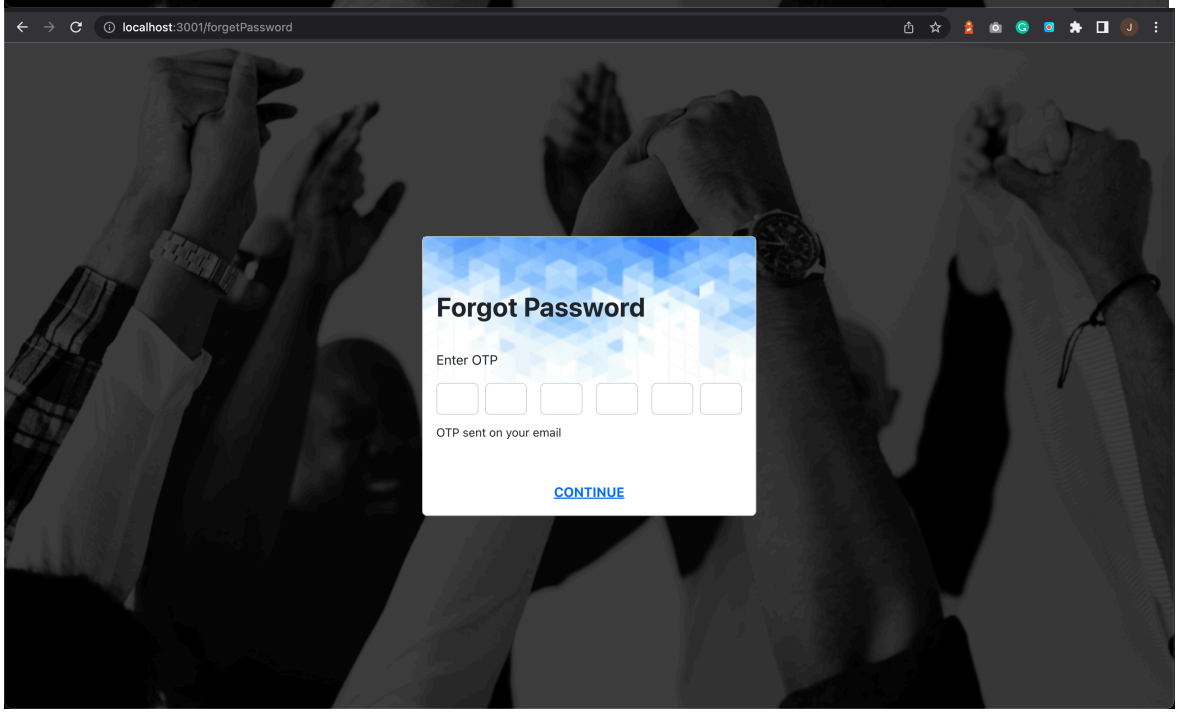
We started working with react js, bootstrap for developing our front-end with responsive screens, we decided to start with role 'employee'. Pages we have achieved so far:

Login:

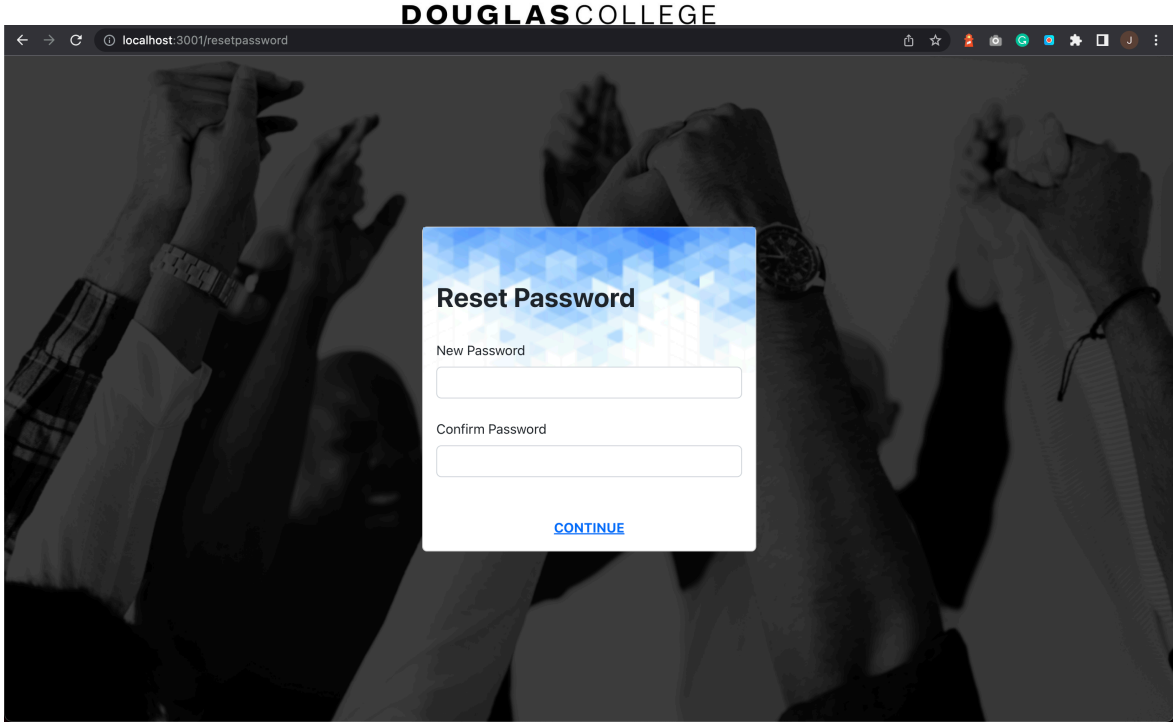
This screen will allow users to login with credentials provided by organization or can reset password using OTP

DOUGLAS COLLEGE

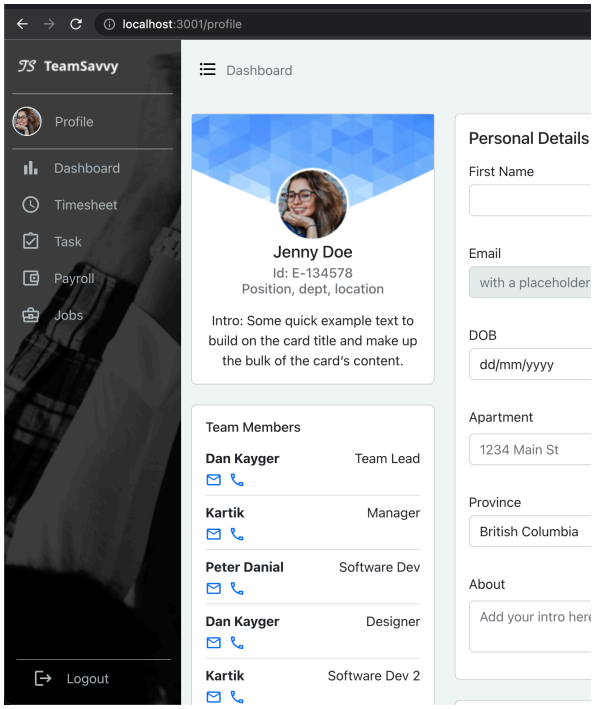




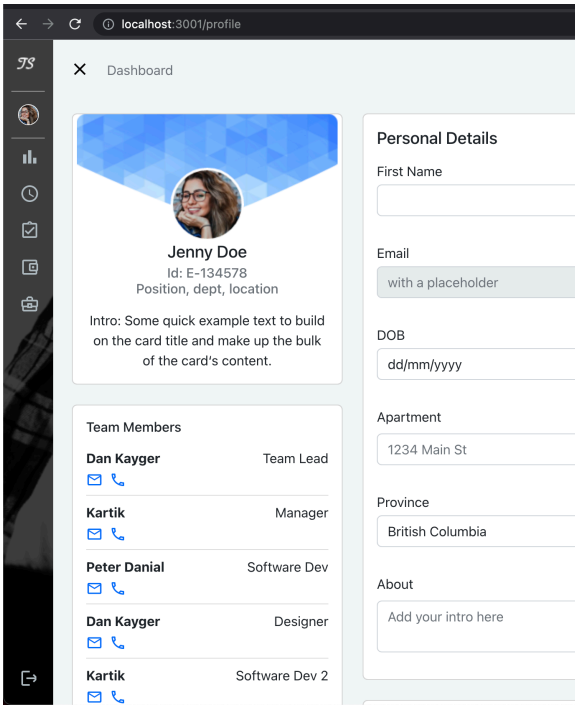
DOUGLAS COLLEGE



Sidenav:
The animated sidebar with navigation links to access the platform.



Open

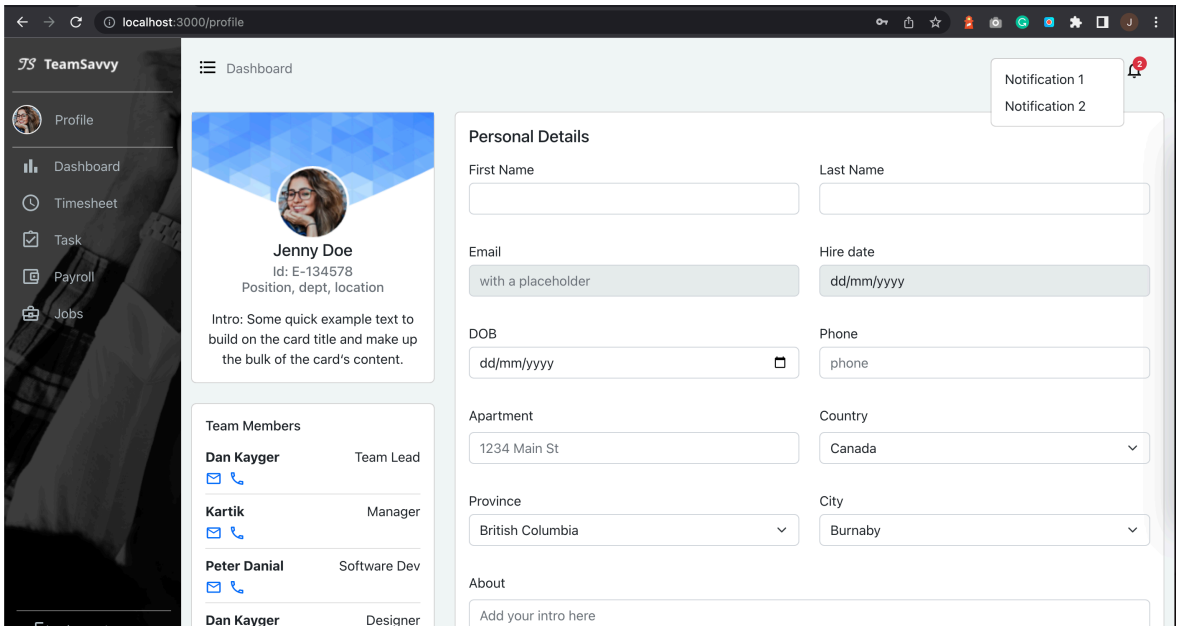


Closed

DOUGLAS COLLEGE

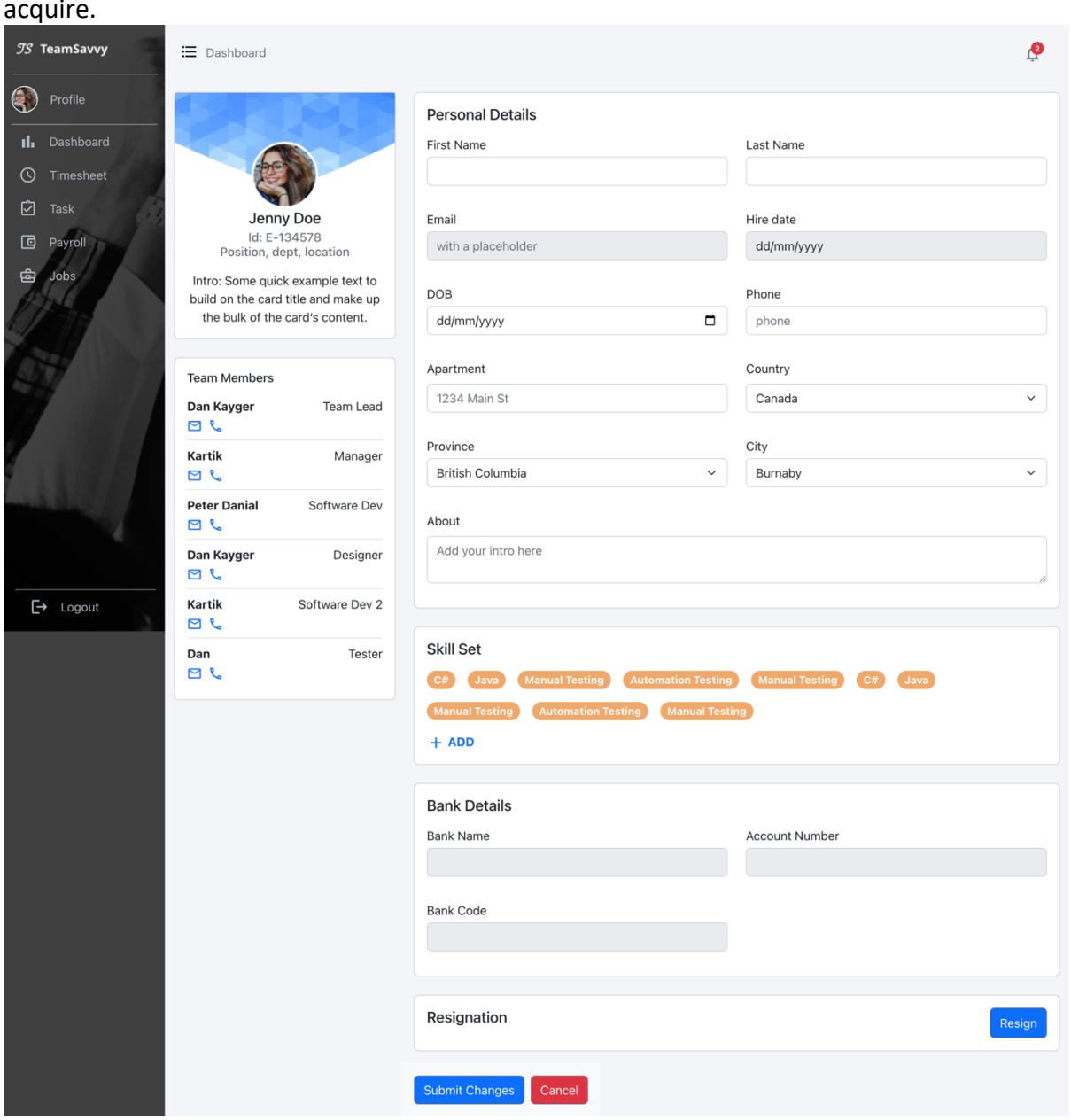
Header and Navigation:

Header has dashboard title which will be breadcrumb for navigation and notification with indicator, it is hardcoded for now. We will work on it after midterm.



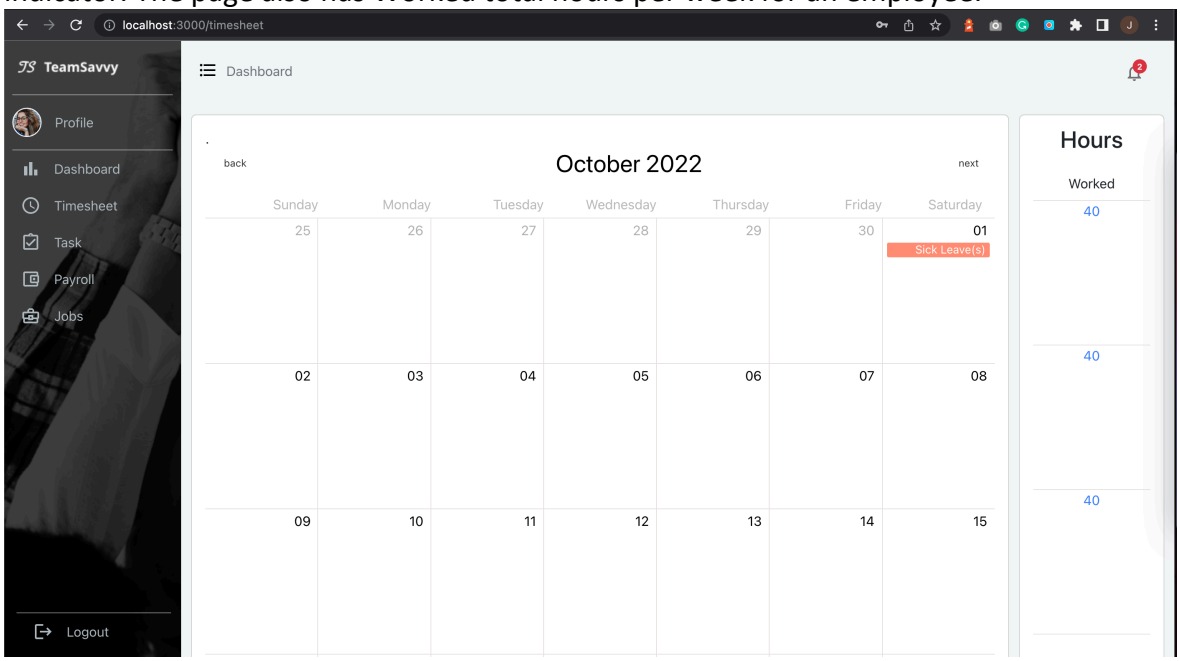
User Profile:

The user profile page has employee information, which is consist of Personal details, Team members with their contact details, Skill set and add new skill option, bank details, and resign option. From here HR or manager can get the overview of an employee and what skills they acquire.



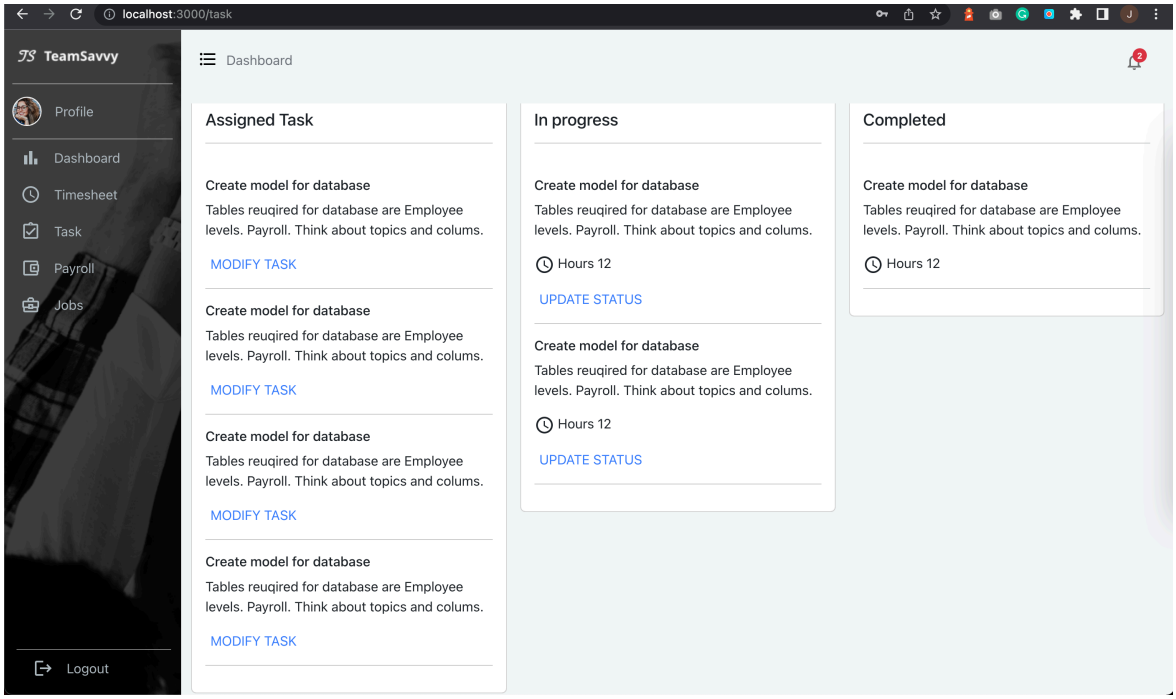
Timesheet:

The Calendar is generated with react-big-calendar which has custom tags for leaves and popups for deleting existing leaves and add new leave. Also have clock-in and clock-out indicator. The page also has Worked total hours per week for an employee.



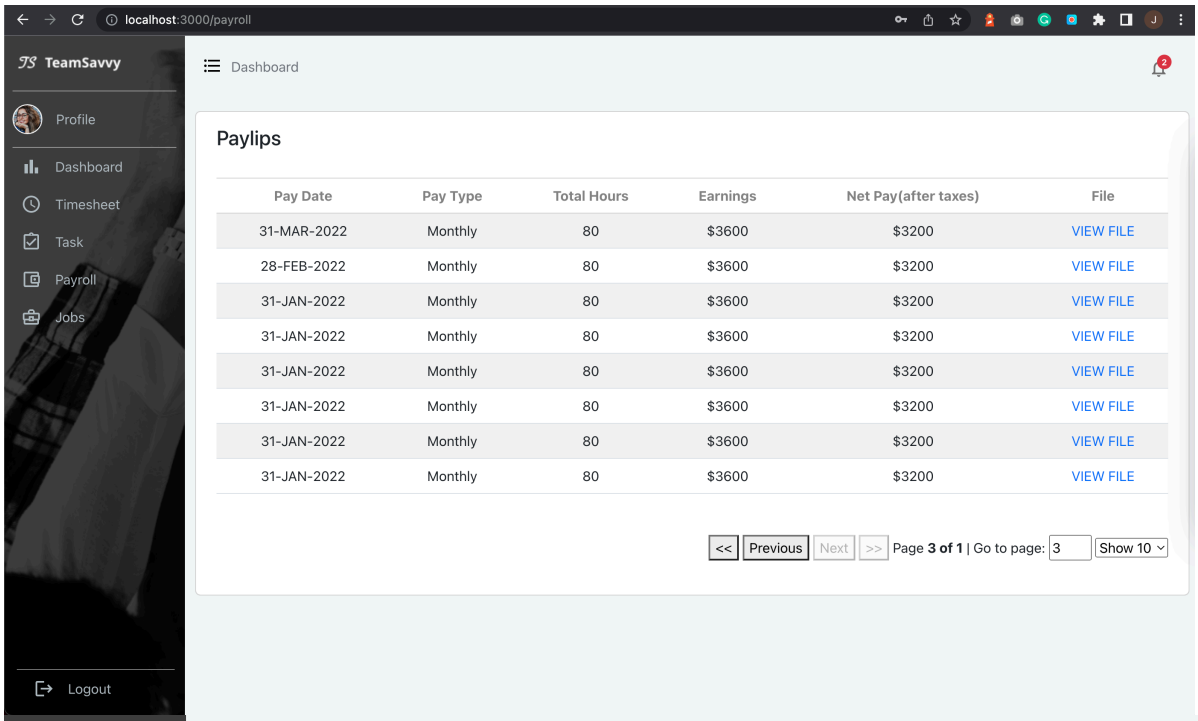
Task:

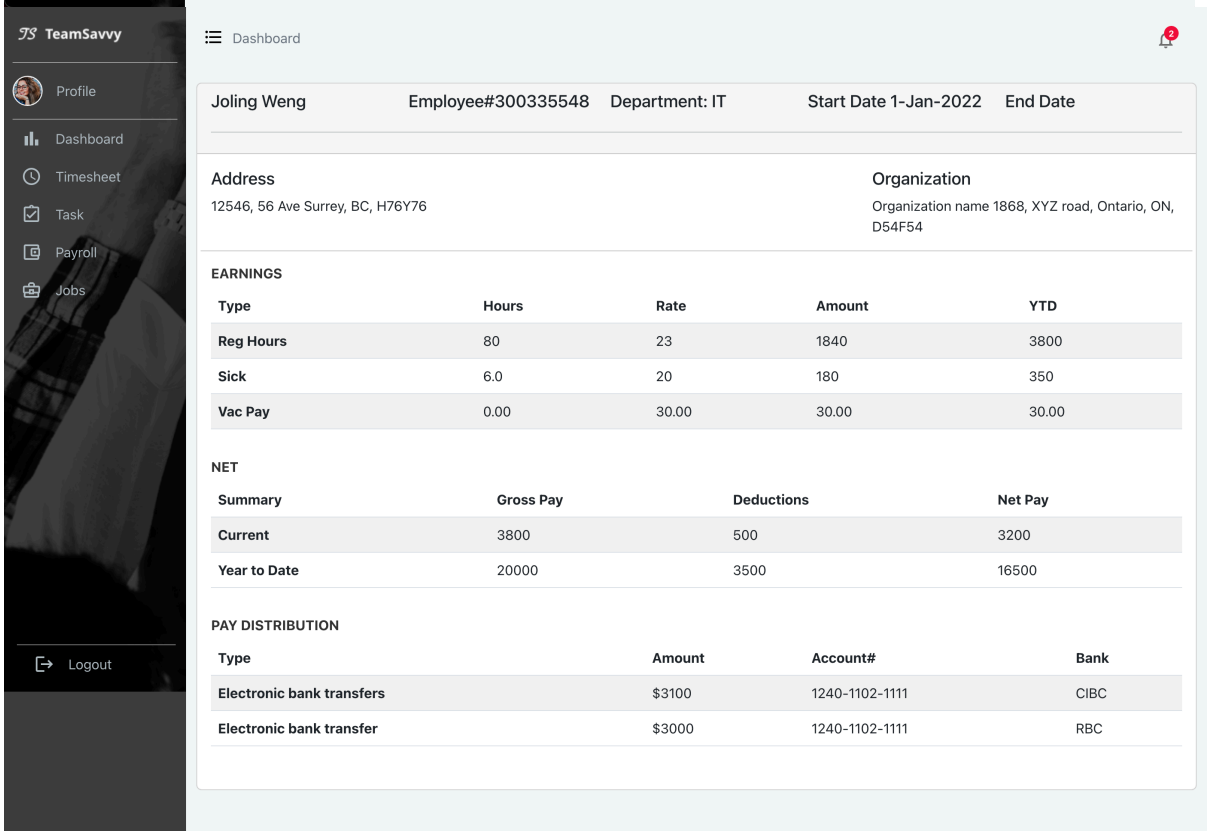
The task page has feature to show assigned, in progress and completed tasks for an employee. We are assuming the employee only working in one project at a time. On click of modify task user will get pop-up to add hours and change status of the task. On update status, the employee can only change status to completed task.



Payroll:

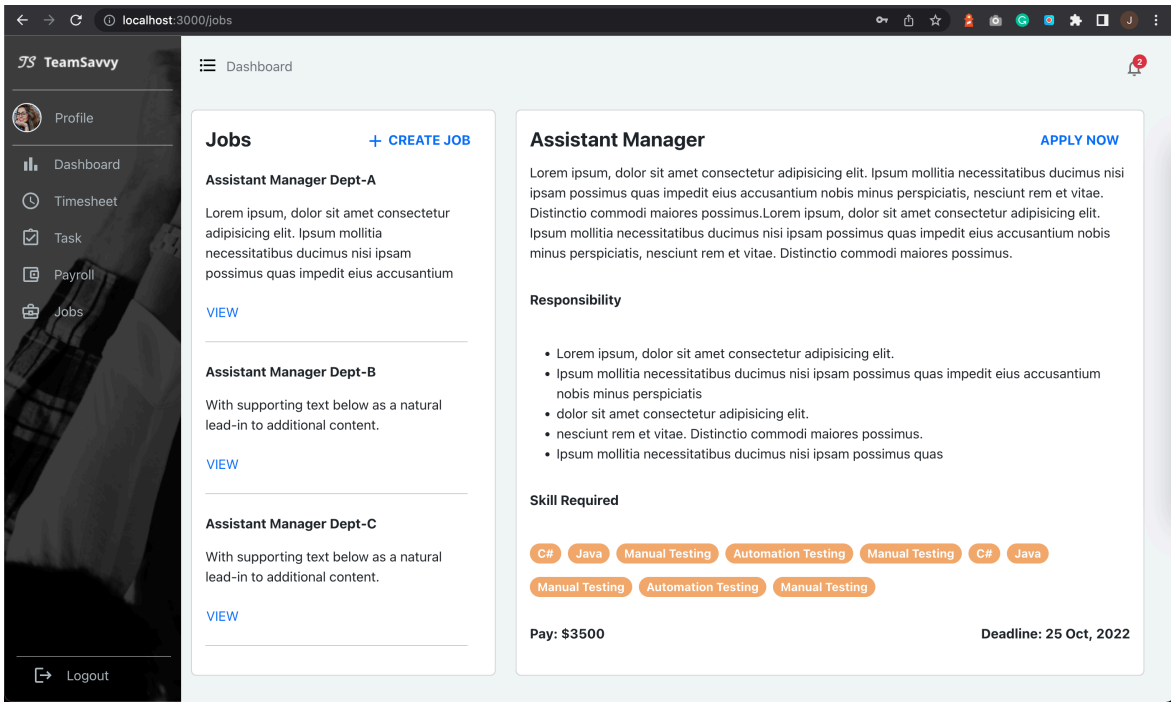
The payroll has payslips screens where employee can check the list of paystubs and also can view the paystubs individually by clicking on view file.





Jobs:

The jobs page for employee have internal jobs created by HR and employee will not see the create job option. Employee can directly apply using Apply now button. HR will get the notification on their portal from where they can check the profile of an employee who applied.



DOUGLAS COLLEGE

Database

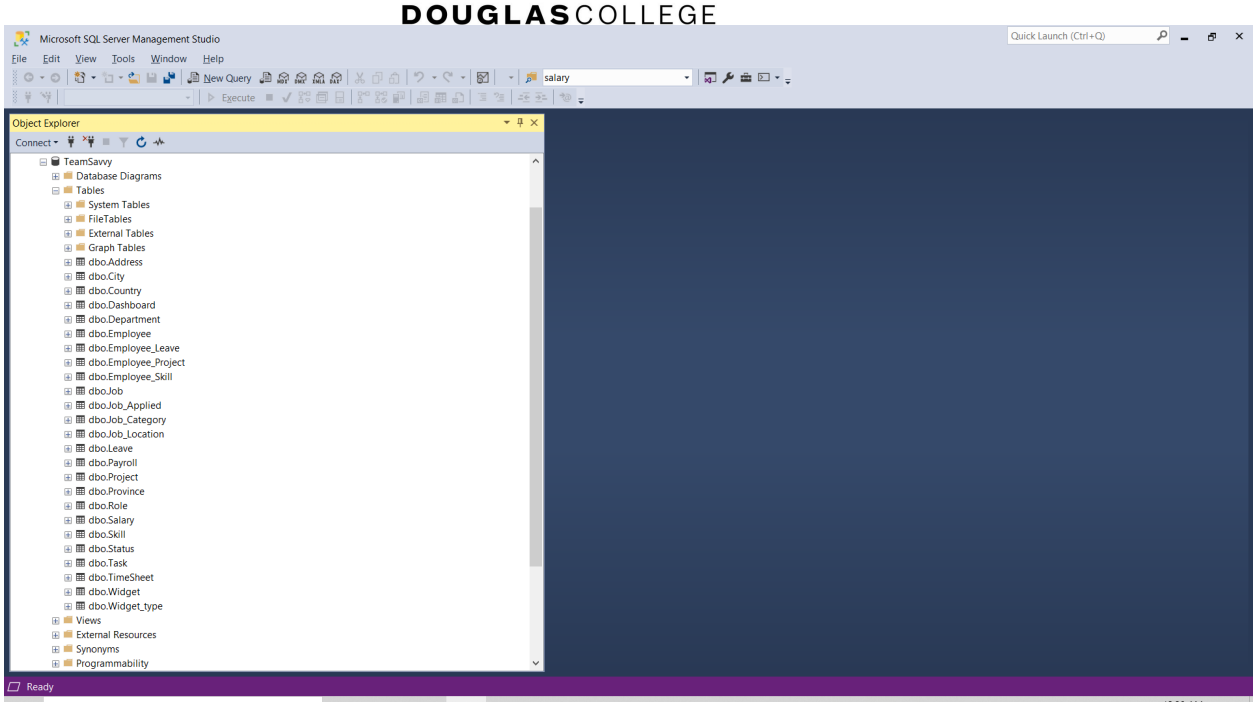
We've created all the Tables that are required for the functioning of the Application. These tables interlinked with each other has all the attributes that gives us the glimpse of how the Application will work.

Structure of the Database:

Tables in our database:

- **Employee**: PK-EmpID, FK – AddressID, DepartmentID, ProjectID, RoleID, StatuesID, First Name, Last Name, DOB, HireDate, Phone, Extension, Email, Bank Account No.
- **Project**: PK- PojectID, FK – Project Manager ID, Project Name, Project Start Date, Project End date, Project Budget, Project Description, Total Task Count, Total Inprogress Count, Total Completed Count, Project Manager, Project Client, Project Load, Project Total Emplyee
- **Leave**: PK – Leave Type ID, FK – Employee ID, Leave Start Date, Leave End Date, Leave Days, Leave Approval Date, Leave Approved By, Leave Status
- **Employee_Leave**: PK – Employee Leave ID, FK – Employee Type ID

- **Employee_Project:** PK – Employee Project ID, FK – Employee ID, Project ID
- **Payroll**: PK – Payroll ID, FK – Employee ID, Pay ID, Pay Date, Pay Type, Total Hours, Earning, Net Pay, Pay_Vacation, Pay_Sick
- **Status**: PK – Status ID, Status_Type
- **Skill**: PK – Skill ID, Skill Name
- **Employee_Skill**: PK – Employee Skill ID,FK – EmployeeID, SkillID
- **Time_Sheet:** PK – TimesheetID, FK – EmployeeID
- **Role**:PK – RoleID, RoleName
- **Department**: PK – DepartmentID, DepartmentName
- **Job:** PK – JobID, FK – JobCategoryID , JobLocationID, DepartmentID, JobRoleID, JobSalary, JobDesc
- **Job Applied**: PK- Job Applied ID, FK – EmployeeID, JobID
- **Job Category:** PK – JobCategoryID, JobCategoryName, JobCategoryValue
- **Job Location**: PK – JobLocationID, FK – CityID, Location, PostalCode
- **Address:** PK – AddressID, FK - CityID, Apartment, Postal Code
- **City**: PK – CityID, CityName, FK – Province ID
- **Province:** PK – Province ID, FK – Country ID, ProvinceName, ProvinceAbbreviations
- **Country**: PK – CountryID, Country Name
- **Dashboard**: PK – DashboardID,FK – EmployeeID, WidgetID
- **Widget**: PK – WidgetID, FK – WidgetTypeID, WidgetName)
- **Widget_Type:** PK – WidgetTypeID, WidgetType
- **Task:** PK – TaskID, FK – EmployeeID, FK-ProjectID, TaskName, TaskDescription, TaskStartDate, TaskEndDate, TaskTotalHours, AssignedBy, AssignedTo, AssignedDate, TaskStatus
- **Salary**: PK – (SalaryID,SalaryIncdate), FK – EmployeeID, SalaryType, Salary



Above screenshot is about TeamSavvy database with all the tables required for application as of now and attaching the SQL script to generate the above database with tables and populate data.

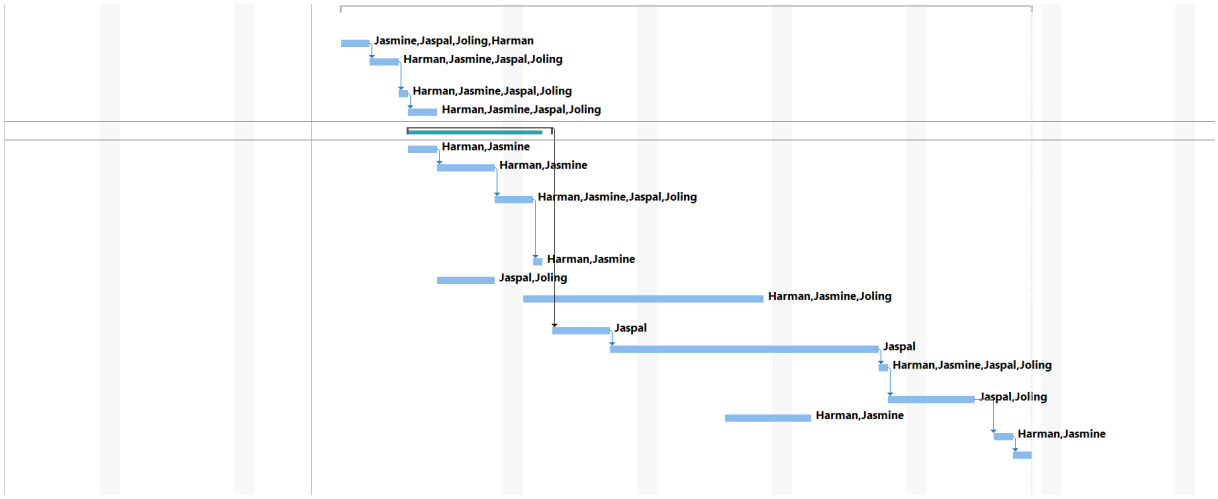
Pending Implementations:

A Gantt chart, a Network Diagram and Timeline on Microsoft Project:

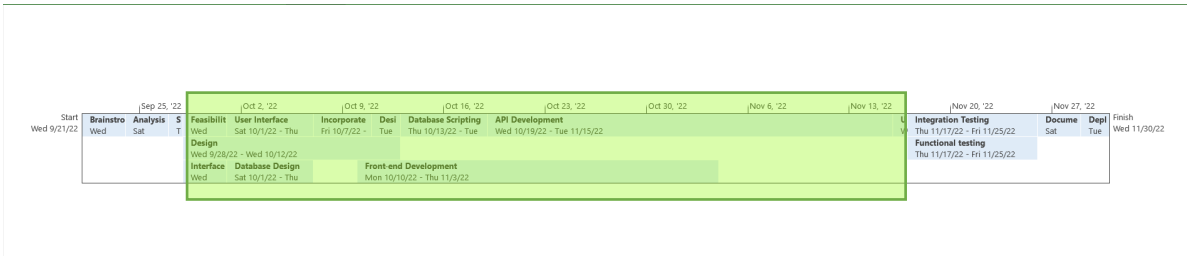
Network Diagram: We have completed the content under the green box so far. Every other task is in progress.

Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	Add New Column
	Software Development	**56 days?**	**Wed 9/21/22**	**Thu 12/1/22**			
	Brainstroming	3 days	Wed 9/21/22	Fri 9/23/22		Jasmine,Jaspal,Joling,Harman	
	Analysis/Software Requirements	3 days	Sat 9/24/22	Mon 9/26/22	1	Harman,Jasmine,Jaspal,Joling	
	Scope Detrmination	1 day	Tue 9/27/22	Tue 9/27/22	2	Harman,Jasmine,Jaspal,Joling	
	Feasibility Analysis	3 days	Wed 9/28/22	Fri 9/30/22	3	Harman,Jasmine,Jaspal,Joling	
	Design	**12 days**	**Wed 9/28/22**	**Wed 10/12/22**			
	Interface Analysis	3 days	Wed 9/28/22	Fri 9/30/22		Harman,Jasmine	
	User Interface design	5 days	Sat 10/1/22	Thu 10/6/22	6	Harman,Jasmine	
	Incorporate feedback into functional specifications	2 days	Fri 10/7/22	Mon 10/10/22	7	Harman,Jasmine,Jaspal,Joling	
	Design complete	1 day	Tue 10/11/22	Tue 10/11/22	8	Harman,Jasmine	
	Database Design	5 days	Sat 10/1/22	Thu 10/6/22		Jaspal,Joling	
	Front-end Development	20 days	Mon 10/10/22	Thu 11/3/22		Harman,Jasmine,Joling	
	Database Scripting	5 days	Thu 10/13/22	Tue 10/18/22	5	Jaspal	
	API Development	20 days	Wed 10/19/22	Tue 11/15/22	12	Jaspal	
	UI and API integration	1 day	Wed 11/16/22	Wed 11/16/22	13	Harman,Jasmine,Jaspal,Joling	
	Integration Testing	7 days	Thu 11/17/22	Fri 11/25/22	14	Jaspal,Joling	
	Functional testing	7 days	Mon 10/31/22	Tue 11/8/22		Harman,Jasmine	
	Documentation	2 days	Mon 11/28/22	Tue 11/29/22	15	Harman,Jasmine	
	Deployment	2 days?	Wed 11/30/22	Thu 12/1/22	17		

Gantt chart:



Timeline: We have completed the content under the green box so far.



Dashboard, Timesheet UI, and logic

- The dashboard requires to use additional chart tools and the data fetching and queries are more complicated. Hence, we need more time on this part.
- Timesheet page shows employee leaves, clock in/ clock out time and, allowing employee to manage leaves. We're still figuring out how to implement the logic and debugging.

Task, Job

- Logic for task page is almost done. There is a status update issue with the connection to API. We will fix the logic before next report.
- For job page, we still haven't implemented the connection for posting new jobs from HR and the logic when the employee clicks on apply now. Also, the applied button in HR view will show the corresponding employees' profile.

Database

- Since we were focusing on the UI and logic, we haven't populated enough data for showing and testing. We will insert more data to show so that the application will look more alive and reasonable.

- Also, while we're doing the logic and connection, we found out that we might need to change the database a little to make the application more user-friendly. We will be keeping optimizing it.

Proposed Revisions:

As shown in the previous report, due to the restriction of device, we will only use mobile app for clock in/ clock out purpose for each employee without saving the biometric information to user database. This is because Android has restricted it and only shares the function for using.