Agile Development In
Cloud Computing Environments

WS 2025-26

WORKFORCE PLANNING TOOL

Masters in Engineering
Information Technology

Team Members

Pavithran Padmanaban	1564590
Babitha Nadar	1566929
Sushmitha Halkurike Pallarappa	1565737
Senthil Arumugam Ramasamy	1565724
Saranya Elumalai	1564318

Introduction

Workforce planning is a critical function in large organizations, as it involves assigning employees to projects based on skills, availability, and organizational priorities. Inefficient or manual workforce allocation often leads to delays, poor resource utilization, and lack of transparency across departments. To overcome these challenges, organizations require structured digital systems that support employee requests, approvals, and assignments. This project focuses on the development of a Workforce Management Tool as part of the Agile Development course. The objective of the project is to design and implement a digital workflow system that supports the internal assignment of employees to projects. The system models real organizational roles such as Project Manager, Department Head, Resource Planner, and Employee, each contributing to different stages of the workforce planning process. The application was developed using an individual technology approach, with a backend based on Java 21, Spring Boot, Gradle, and a microservice oriented architecture, and a frontend implemented using React and JavaScript. MongoDB was used as the database to manage flexible employee and project data, and the application was deployed on a cloud platform. Agile principles, including SCRUM and Kanban, were applied throughout the project to support iterative development, task tracking, and continuous validation. This report documents the system design, agile process, and key insights gained during the project.

Agile Methodology (Scrum / Kanban):

Agile methodology is a modern approach to software development that focuses on iterative progress, flexibility, and close collaboration between team members and stakeholders. Rather than following a rigid, sequential process, Agile encourages continuous planning, development, testing, and improvement throughout the project lifecycle. In this project, Agile methodology was applied to manage the development of the Workforce Management Tool, which involved evolving requirements and complex workflows. The team adopted Scrum as the primary framework to structure development activities, supported by Kanban practices for task tracking and workflow visualization. This combination allowed the team to break down requirements into user stories, plan work in short iterations, and continuously validate progress while maintaining transparency and adaptability.

The Agile methodology is guided by a set of core principles that were actively applied during the project:

a. Customer and stakeholder collaboration: Requirements and expectations were continuously reviewed to ensure alignment between the implemented solution and business needs.
b. Responding to change: As the understanding of workforce workflows evolved, the team adapted data models, APIs, and system design without disrupting overall progress.
c. Frequent delivery of working software: Functional features such as user management, project creation, and employee handling were implemented and validated incrementally.
d. Emphasis on individuals and interactions: Effective communication and collaboration within the team played a crucial role in resolving technical and design challenges.

e. Continuous improvement: Continuous integration and delivery are essential practices in Agile development, ensuring that software changes are frequently integrated, tested, and kept in a deployable state. This approach improves software quality and reduces risks by enabling early detection of issues and smoother deployment processes.

f. Review and Process Improvement: Agile promotes regular review sessions, such as retrospectives, which allow the team to evaluate its performance and identify opportunities for improvement. This continuous feedback process supports ongoing refinement of development practices and enables the team to adapt effectively throughout the project lifecycle.

Kanban for Task Management and Workflow Visualization:

Kanban was used alongside SCRUM to support visual task management and continuous workflow during the development of the Workforce Management Tool. The Kanban method focuses on visualizing work, limiting work in progress, and optimizing the flow of tasks from start to completion. By representing tasks on a Kanban board and organizing them into stages such as "To Do," "In Progress," and "Done," the team achieved better transparency and awareness of project status at any given time.

The use of Kanban enabled the application of the pull principle, where team members selected tasks based on their availability and expertise rather than having tasks assigned to them. This approach helped balance workload, and improve individual ownership of tasks. Kanban also allowed the team to manage changes effectively, as new tasks or refinements could be added to the board without disrupting ongoing work. Overall, the integration of Kanban practices complemented SCRUM by providing a flexible and efficient mechanism for tracking progress and supporting continuous delivery throughout the project lifecycle.

SCRUM as an Agile Framework:

SCRUM was adopted as the primary Agile framework for the development of the Workforce Management Tool because it provides a structured yet flexible approach to managing complex software projects. As an Agile framework, SCRUM emphasizes iterative development, transparency, and continuous feedback, which were essential for handling evolving requirements and understanding the workforce planning workflows defined for the project. Instead of delivering the system as a single final product, SCRUM enabled the team to incrementally build and validate functionality in smaller, manageable units. In this project, SCRUM principles were applied through the use of user stories, sprint planning, and incremental task completion. User stories were derived from system roles such as Project Manager, Department Head, and Employee, ensuring that development activities remained focused on delivering business value. Sprint planning helped the team organize work logically and prioritize core functionalities such as user creation, project management, and employee handling. Regular evaluation of progress ensured transparency and allowed early identification of technical or conceptual challenges. By applying SCRUM as an Agile framework, the team was able to maintain adaptability, improve collaboration, and deliver working software incrementally. This approach supported effective coordination between backend, frontend, and database development, and ensured that the final solution aligned closely with the intended workforce management objectives.

System Architecture (High Level):

The Workforce Management Tool was designed using a high level microservice based architecture (Figure 1) to ensure scalability, modularity, and clear separation of concerns. The system is composed of three primary layers: the frontend layer, the backend services layer, and the database layer. Each layer is responsible for a specific set of functionalities and communicates with other layers through well defined interfaces, primarily RESTful APIs. This architectural approach supports independent development, testing, and deployment of system components, which aligns well with Agile development principles.

The frontend layer is implemented using React and JavaScript and provides the user interface for interacting with the system. It enables users such as Project Managers, Department Heads, and Employees to create projects, manage profiles, and view assignments. The frontend communicates with backend services through HTTP based REST APIs, ensuring a clear separation between presentation logic and business logic. The backend layer is developed using Java 21, Spring Boot 3.4.0, and Gradle, following a microservice oriented design. Each backend service handles a specific domain, such as user management, project management, or employee data handling. This design improves maintainability and allows services to evolve independently as system requirements change. Business logic, validation, and workflow handling are implemented within these services, ensuring consistent processing of requests from the frontend.

For deployment, the application is hosted on Render, leveraging its managed cloud services. The React frontend is deployed as a Render Static Site, which serves pre-built static assets efficiently with automatic HTTPS and continuous deployment from the Git repository. The Spring Boot backend services are deployed as Render Web Services, running in a managed containerized environment that supports environment variables, automated builds, and scalable runtime infrastructure. This deployment approach enables seamless integration between frontend and backend, and ensures reliable cloud-native operation without the need for manual server management.

The database layer uses MongoDB, to store and manage application data. MongoDB was chosen for its flexibility in handling complex and nested data structures, such as employee skills, work experience, and project requirements. All data created or modified through the backend APIs is persisted in MongoDB and verified to ensure data integrity. Overall, this high level architecture supports a cloud ready, scalable, and extensible workforce management system suitable for enterprise environments.

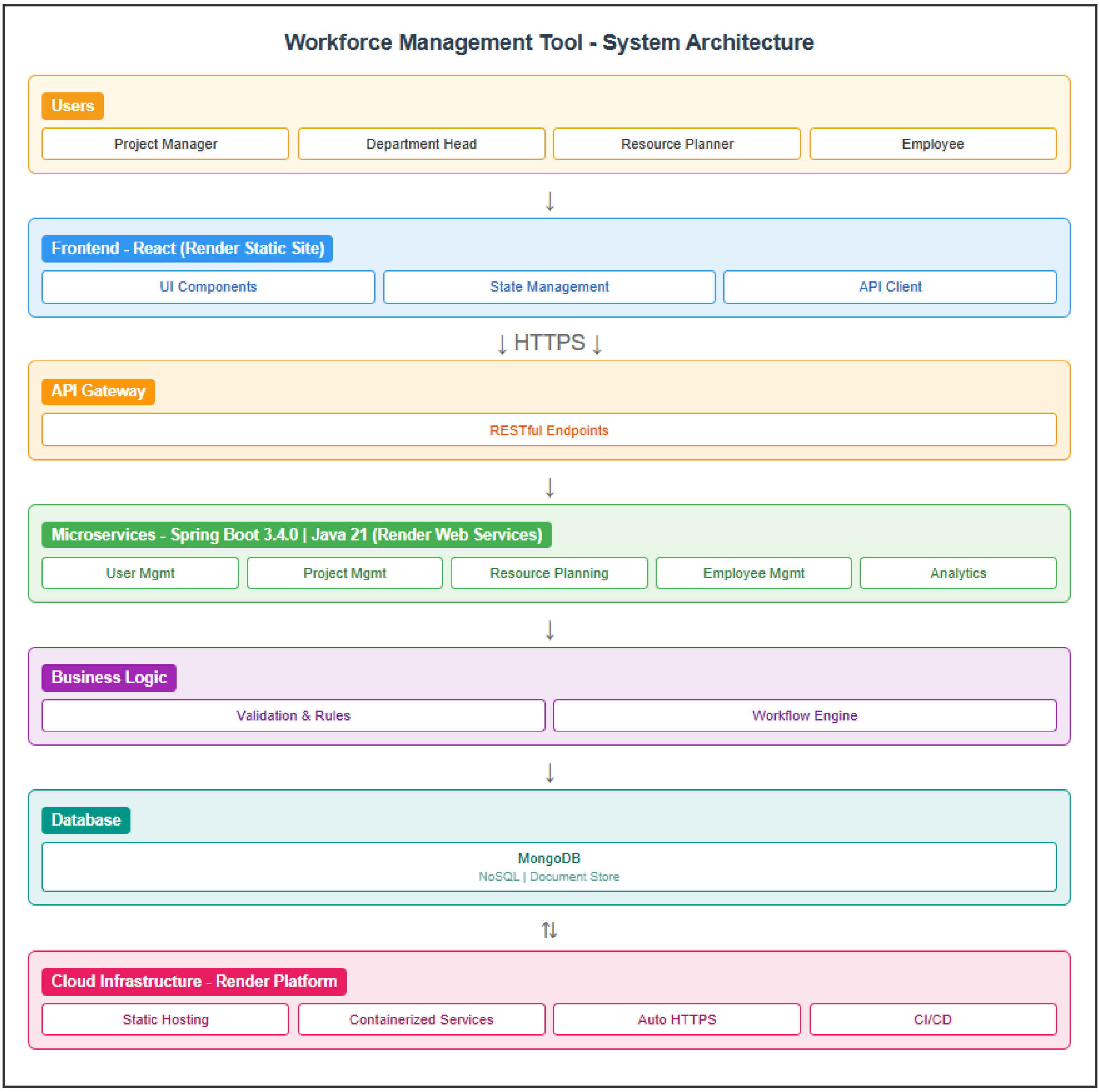
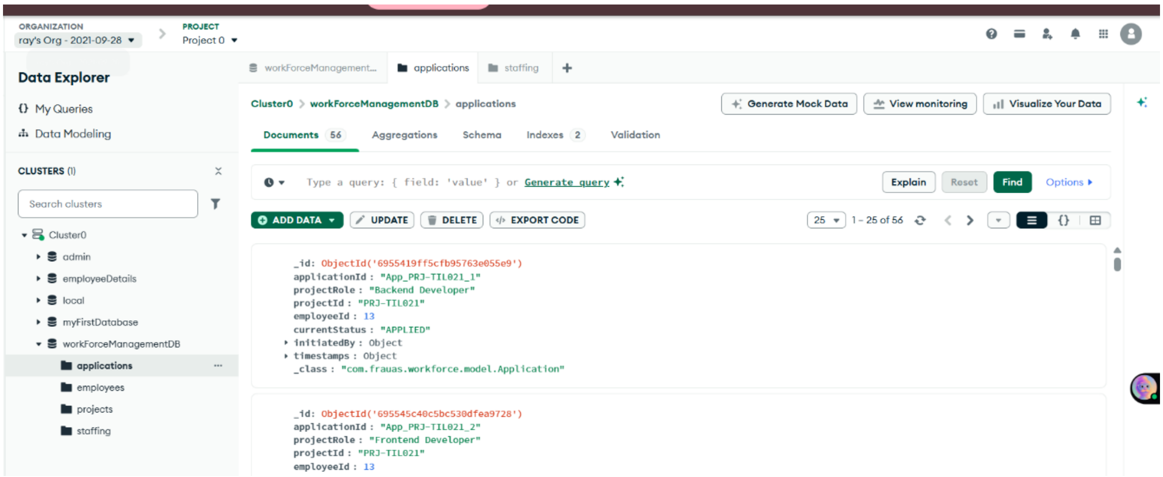


Figure 1: System Architecture

Database Design (MongoDB):

The application uses MongoDB Atlas (Figure 2), a cloud-hosted NoSQL document database, to store and manage workforce-related data such as employees, projects, and assignments in a flexible and scalable manner. MongoDB is integrated into the backend using Spring Data MongoDB through the spring-boot-starter-data-mongodb dependency, enabling seamless repository based data access with automatic index creation for efficient querying. The system connects securely to the MongoDB Atlas cluster using a connection string configured with retryable writes and a majority write concern to ensure data reliability and consistency. The database design follows an employee based master data model as defined in the project requirements, where each employee is stored as a document containing personal details, department information, skills, interests, availability, work experience, and assignment status. This document-oriented structure allows rich employee profiles while remaining extensible for future enhancements. Project data is stored in separate documents that capture project descriptions, timelines, required roles, competencies, and capacity details, with embedded documents and arrays used to efficiently represent role requirements and skill mappings within each project.



```
_id: ObjectId('6955419ff5cfb95763e055e9')
applicationId : "App_PRJ-TIL021_1"
projectRole : "Backend Developer"
projectId : "PRJ-TIL021"
employeeId : 13
currentStatus : "APPLIED"
▸ initiatedBy : Object
▸ timestamps : Object
_class : "com.frauas.workforce.model.Application"

_id: ObjectId('695545c40c5bc530dfea9728')
applicationId : "App_PRJ-TIL021_2"
projectRole : "Frontend Developer"
projectId : "PRJ-TIL021"
employeeId : 13
```

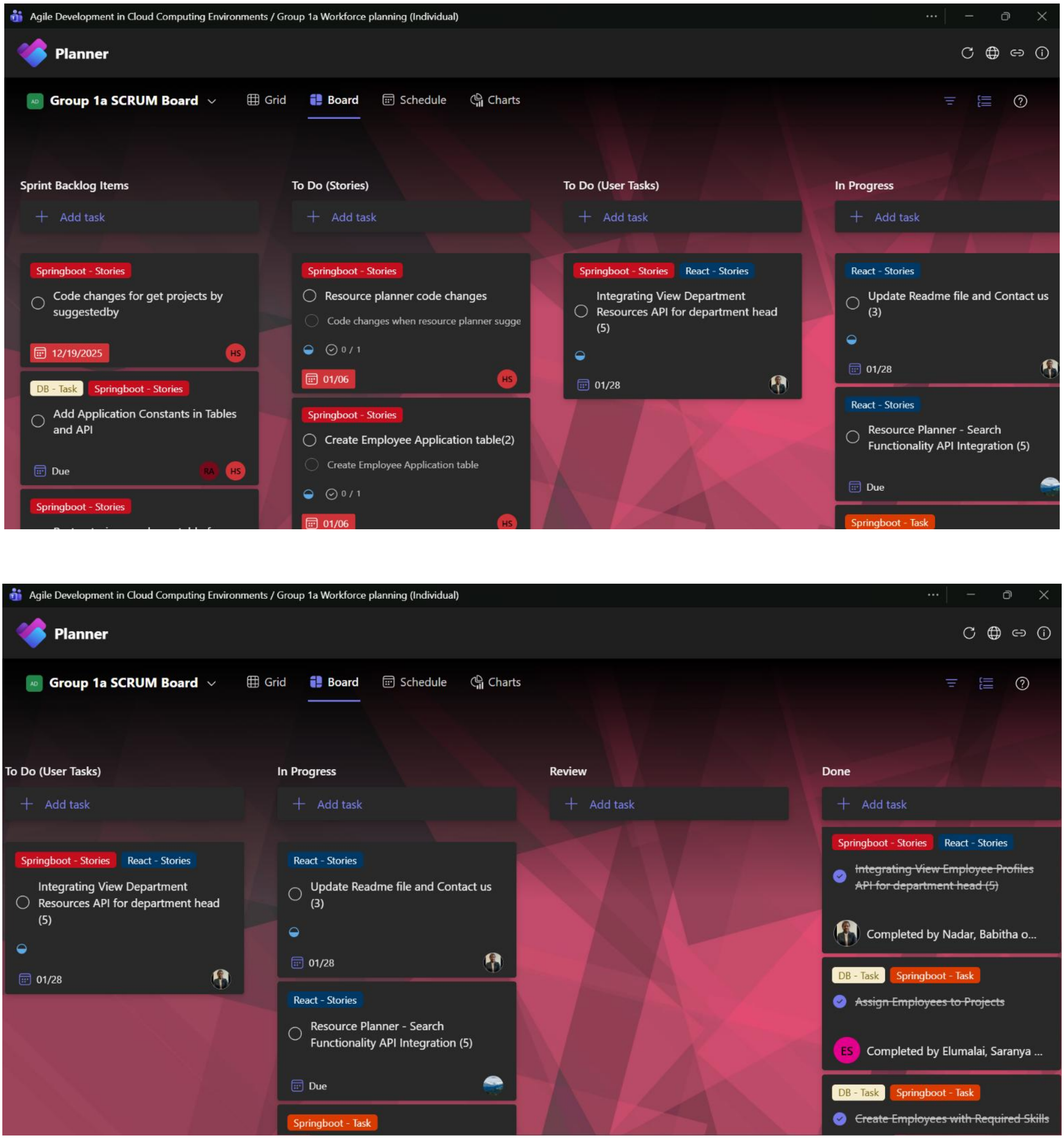
Figure 2: MongoDB

Sprint Planning:

Sprint planning was used to organize the development of the Workforce Management Tool into structured and manageable iterations. At the beginning of each sprint, the team identified relevant user stories, defined tasks, and prioritized functionality based on project goals. This approach ensured steady progress and allowed continuous evaluation through Scrum meetings, Sprint planning, Sprint reviews and retrospectives, supporting iterative improvement throughout the project.

Kanban Boards:

A Kanban board (Figure 3) was used to visualize and manage tasks during the development process. Tasks were organized into stages such as “Backlog”, “To Do,” “In Progress,” and “Done,” enabling clear visibility of work status. The pull based approach allowed team members to select tasks according to availability, which helped balance workload and improve transparency across the team.


Agile Development in Cloud Computing Environments / Group 1a Workforce planning (Individual)
Planner
Group 1a SCRUM Board
Grid
Board
Schedule
Charts

Sprint Backlog Items

- Springboot - Stories — Code changes for get projects by suggestedby — 12/19/2025 — HS
- DB - Task, Springboot - Stories — Add Application Constants in Tables and API — Due — RA, HS
- Springboot - Stories

To Do (Stories)

- Springboot - Stories — Resource planner code changes — Code changes when resource planner sugge — 0 / 1 — 01/06 — HS
- Springboot - Stories — Create Employee Application table(2) — Create Employee Application table — 0 / 1 — 01/06 — HS

To Do (User Tasks)

- Springboot - Stories, React - Stories — Integrating View Department Resources API for department head (5) — 01/28

In Progress

- React - Stories — Update Readme file and Contact us (3) — 01/28
- React - Stories — Resource Planner - Search Functionality API Integration (5) — Due
- Springboot - Task

Review

Done

- Springboot - Stories, React - Stories — ~~Integrating View Employee Profiles API for department head (5)~~ — Completed by Nadar, Babitha o...
- DB - Task, Springboot - Task — ~~Assign Employees to Projects~~ — ES Completed by Elumalai, Saranya ...
- DB - Task, Springboot - Task — ~~Create Employees with Required Skills~~

Figure 3: Kanban Boards

- **Sprint 1 (4 weeks: 31.10.2025 – 28.11.2025):**

During the first sprint, the team focused on establishing the foundational structure of the Workforce Management Tool based on the provided requirements. The sprint began with understanding the overall system objectives and translating them into initial user stories, with core roles such as Project Manager, Department Head, Resource Planner, and Employee being identified and modeled. The development effort started with Project Manager functionality, as this role initiates the workforce planning process. Backend setup activities included initializing the Spring Boot project, configuring required dependencies, and integrating MongoDB for data persistence. Initial REST API endpoints were implemented for user creation, role management, and basic project-related operations, and these APIs were tested using Swagger. Data persistence and correctness were verified directly in MongoDB to ensure reliable storage of user and role information. This sprint established a stable technical and functional foundation for future development, followed by a sprint review to validate implemented features and a retrospective to identify challenges and improve planning for subsequent sprints.

- **Sprint 2 (3 weeks: 28.11.2025 – 19.12.2025):**

During Sprint 2, the team focused on expanding the core functionality of the system by implementing key workforce planning features. The remaining Project Manager functionalities were completed, including full support for project creation with detailed project information such as required roles, skills, capacity, timelines, and locations. In parallel, Employee-related functionality was implemented, enabling the creation and management of employee master data, including skills, availability, interests, and work experience. This sprint also marked the beginning of Resource Planner functionality, laying the groundwork for employee search and skill-based matching. Backend REST APIs were extended to support these features, while corresponding frontend components were developed to ensure seamless interaction between the user interface and backend services. MongoDB collections were structured and updated to persist project and employee data consistently. Continuous testing through API validation and database checks ensured data integrity across the system. The sprint review confirmed that the workforce planning features were functional, and the retrospective helped the team improve task estimation accuracy and API validation practices for subsequent sprints.

- **Sprint 3 (4 weeks: 19.12.2025 – 16.01.2026):**

During Sprint 3, the team concentrated on system integration and workflow refinement across all major modules. The primary focus was on integrating frontend and backend components to ensure smooth and reliable communication through REST APIs. During this sprint, the team completed the integration of Project Manager, Employee, and Resource Planner functionalities, ensuring that workflows such as project creation, employee application, skill-based suggestion, and assignment preparation functioned cohesively. At the same time, development of Department Head functionality was initiated to support approval and governance workflows. Workforce related processes such as employee availability handling, data validation, and assignment readiness were refined to align system behavior with defined business logic. Technical issues related to API responses, data formats, and validation errors were identified and resolved, and MongoDB records were consistently verified to ensure correct persistence of updated data. Usability improvements were made on the frontend, and backend error handling was strengthened based on feedback from earlier sprints. The sprint review validated the end-to-end functionality of integrated workflows, while the retrospective highlighted improved collaboration and communication across the team.

- **Final Phase (2 weeks: 16.01.2026 – 28.01.2026):**

  The final phase of the project focused on Integration, verification, and preparation for submission. During this phase, the team concentrated on resolving remaining issues, improving data consistency, and validating all core functionalities. Extensive verification was performed to ensure that users, projects, and employee data were correctly stored and retrievable from MongoDB. Minor refinements were made to the frontend and backend to enhance usability and reliability. Documentation activities were carried out in parallel, including preparation of the project report and consolidation of project artifacts. The final phase ensured that the system met all defined requirements and was ready for evaluation and submission.

Scrum Meetings:

Scrum meetings were an integral part of managing and coordinating the development of the Workforce Management Tool. These meetings provided a structured way for the team to plan work, monitor progress, and continuously evaluate development activities. Regular meetings supported transparency and ensured that all team members shared a common understanding of sprint goals, ongoing tasks, and potential challenges. By following Scrum meeting practices, the team was able to maintain alignment throughout the project lifecycle and respond effectively to changes or issues that occurs during development.

Scrum Roles:

Within the Scrum framework, specific roles were followed to ensure effective collaboration and accountability. The Scrum Master role focused on facilitating Scrum practices, project requirements and objectives, which guided prioritization of user stories and features, supporting communication, documentation and helping the team overcome impediments. The Development Team consisted of members responsible for backend, frontend, and database development, who worked collaboratively to implement and integrate system functionality. This role distribution enabled efficient coordination while supporting cross functional teamwork.

Sprint Review:

Scrum reviews were conducted at the end of each sprint to evaluate completed work and assess whether sprint objectives had been met. During these reviews, implemented features such as user creation, project setup, employee management, and data persistence were examined and validated. Feedback gathered during the review process was used to identify improvements, refine workflows, and address technical issues. Scrum reviews played a key role in ensuring continuous improvement and alignment of the developed system with overall project goals.

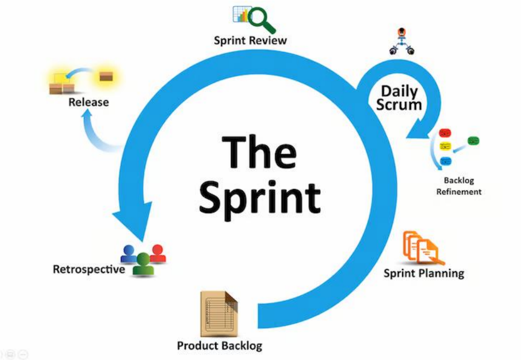


Figure 4: Sprint Overview

Critical Appraisal on Managing SCRUM:

(What was good? What went bad? Challenges?)

One of the strongest aspects of applying SCRUM in the Workforce Management Tool project was the clear structuring of work through sprints, user stories, and a Kanban board. Dividing the project into multiple sprints, as shown in the sprint timeline, enabled incremental development and continuous validation of results. User stories derived from the documented use cases helped maintain a strong connection between requirements and implementation. The use of a Kanban board supported transparency and allowed team members to follow the pull principle, selecting tasks based on availability and expertise. This approach improved collaboration across backend, frontend, and database development and ensured steady progress throughout the project lifecycle.

Despite these positive outcomes, some aspects of managing SCRUM did not work as effectively as expected. Effort estimation during sprint planning proved challenging, as several tasks were more complex than initially anticipated. In particular, tasks involving MongoDB data modeling, validation logic, and REST API integration required additional time and iterations. As a result, certain tasks extended beyond their planned sprint and had to be reprioritized. Additionally, SCRUM ceremonies such as retrospectives were conducted in a simplified manner due to academic time constraints, which limited the depth of process reflection compared to a full industrial SCRUM environment.

The project also presented several challenges that required adaptive problem solving. Translating high level workforce planning workflows into concrete technical implementations demanded multiple refinements and design adjustments. Coordinating changes across multiple system layers, including frontend components, backend services, and database structures, required careful communication and validation. Learning and integrating multiple technologies simultaneously further increased complexity. However, these challenges reinforced the importance of flexibility, continuous improvement, and adaptation of SCRUM practices. Overall, the experience demonstrated that SCRUM, when applied thoughtfully and critically, is effective even in complex academic projects with limited time and evolving requirements.

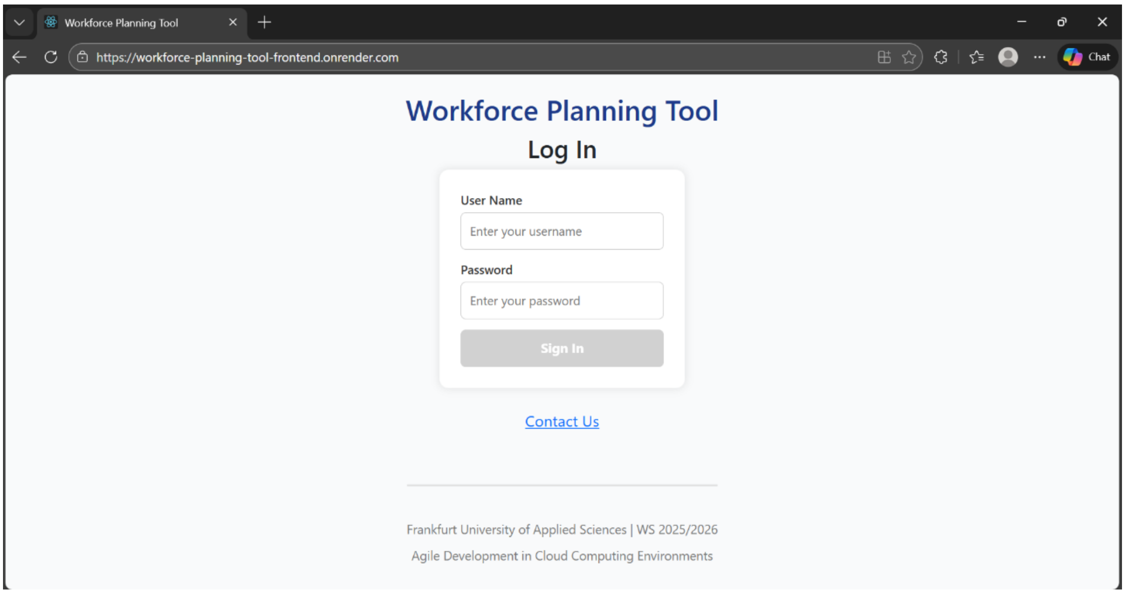
Workforce Planning Tool (WPT)

Workforce Planning Tool: https://workforce-planning-tool-frontend.onrender.com/

Login Page – Workforce Planning Tool (WPT):

The login page (Figure 5) serves as the entry point to the Workforce Planning Tool and provides controlled access to the system based on user credentials. It enables authenticated access for different organizational roles such as Project Managers, Department Heads, Resource Planners, and Employees, as defined in the project requirements. By enforcing authentication at the initial stage, the system ensures that only authorized users can interact with workforce planning functionalities.

The interface consists of input fields for username and password, along with a sign-in action that initiates authentication against the backend services. The login process is integrated with the backend application, which validates user credentials and determines role based access after successful authentication. This mechanism supports the requirement for role specific workflows, ensuring that users are directed to appropriate dashboards and functionalities based on their assigned roles. Additionally, the login page provides system related information for users in case of connectivity issues, emphasizing system reliability and maintainability.



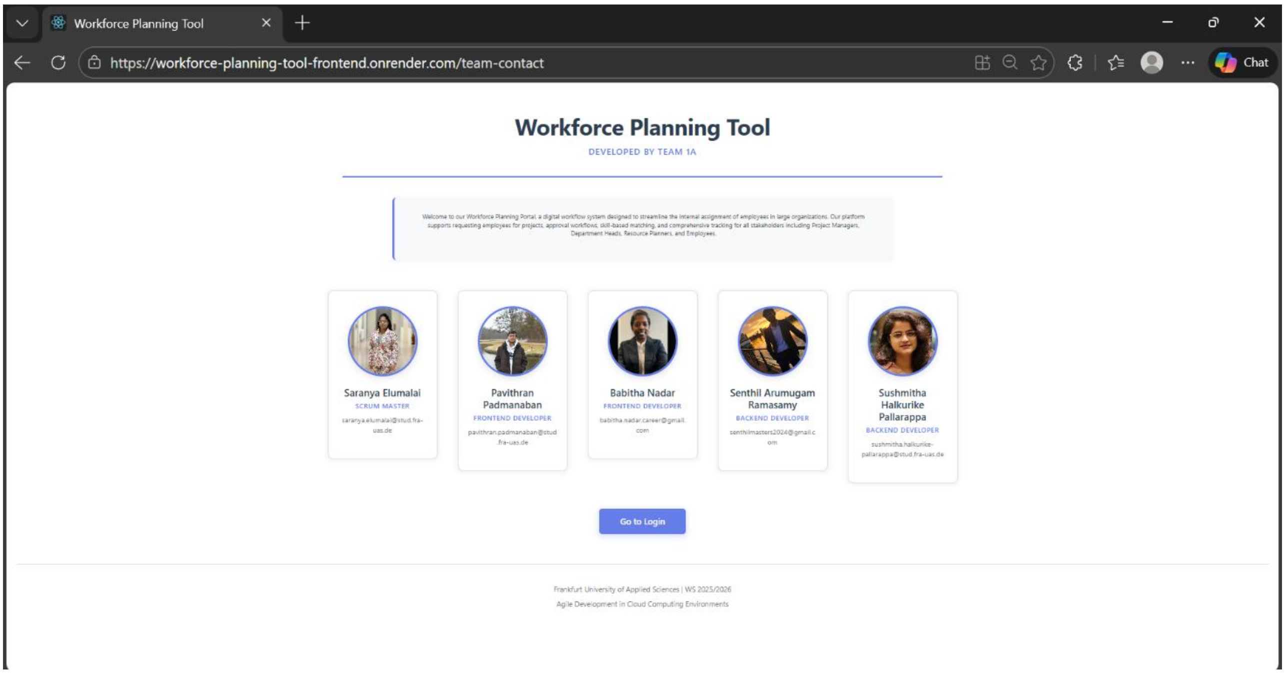
Frankfurt University of Applied Sciences | WS 2025/2026

Agile Development in Cloud Computing Environments

Figure 5: Login Page

Contact Page:

The Contact page provides an informational and supportive interface within the Workforce Planning Tool, presenting details about the development team responsible for the application. This page serves as a reference point for users who require assistance, clarification, or technical support related to the system. The page displays team members in a structured layout, showing each member's name, role within the project (such as Scrum Master, Frontend Developer, or Backend Developer), and contact information. This information helps users understand the responsibilities and expertise areas of the team members. An important interactive feature of this page is the profile detail view, which becomes visible when a user clicks on a team member's profile card. This interaction provides additional information such as a brief professional description and areas of expertise. By providing deeper insights into individual contributions, this feature allows users to identify appropriate points of contact based on the type of support or information needed. The Go to Login action at the bottom of the page enables smooth navigation back to the authentication flow, ensuring that this informational page remains well integrated into the application.



Workforce Planning Tool
https://workforce-planning-tool-frontend.onrender.com/team-contact
Chat

Workforce Planning Tool

DEVELOPED BY TEAM 1A

Welcome to our Workforce Planning Portal, a digital workflow system designed to streamline the internal assignment of employees in large organizations. Our platform supports requesting employees for projects, approval workflows, skill-based matching, and comprehensive tracking for all stakeholders including Project Managers, Department Heads, Resource Planners, and Employees.

Saranya Elumalai
SCRUM MASTER
saranya.elumalai@stud.fra-uas.de

Pavithran Padmanaban
FRONTEND DEVELOPER
pavithran.padmanaban@stud.fra-uas.de

Babitha Nadar
FRONTEND DEVELOPER
babitha.nadar.career@gmail.com

Senthil Arumugam Ramasamy
BACKEND DEVELOPER
senthilmasters2024@gmail.com

Sushmitha Halkurike Pallarappa
BACKEND DEVELOPER
sushmitha.halkurike-pallarappa@stud.fra-uas.de

Frankfurt University of Applied Sciences | WS 2025/2026
Agile Development in Cloud Computing Environments

Figure 6: Contact Us Page

Dashboards – Workforce Planning Tool

User Profile Menu:

The User Profile Menu is available across all dashboards once a user successfully logs into the Workforce Planning Tool which is shown in Figure 7. It is displayed in the top right corner of the interface and provides quick access to essential user related actions, independent of the user's role (Project Manager, Employee, Department Head, or Resource Planner). The profile menu displays the logged in user's name, role, and registered email address within the system. From this menu, users can access Settings to manage their personal profile information or select Logout to securely end their session. The settings screen captures detailed employee information, including personal details (name, email, emergency contact), organizational attributes (department, position, supervisor, assigned project), and location preferences. In addition, users can manage professional attributes such as skills, interests, availability status, contract type, capacity, and work experience.

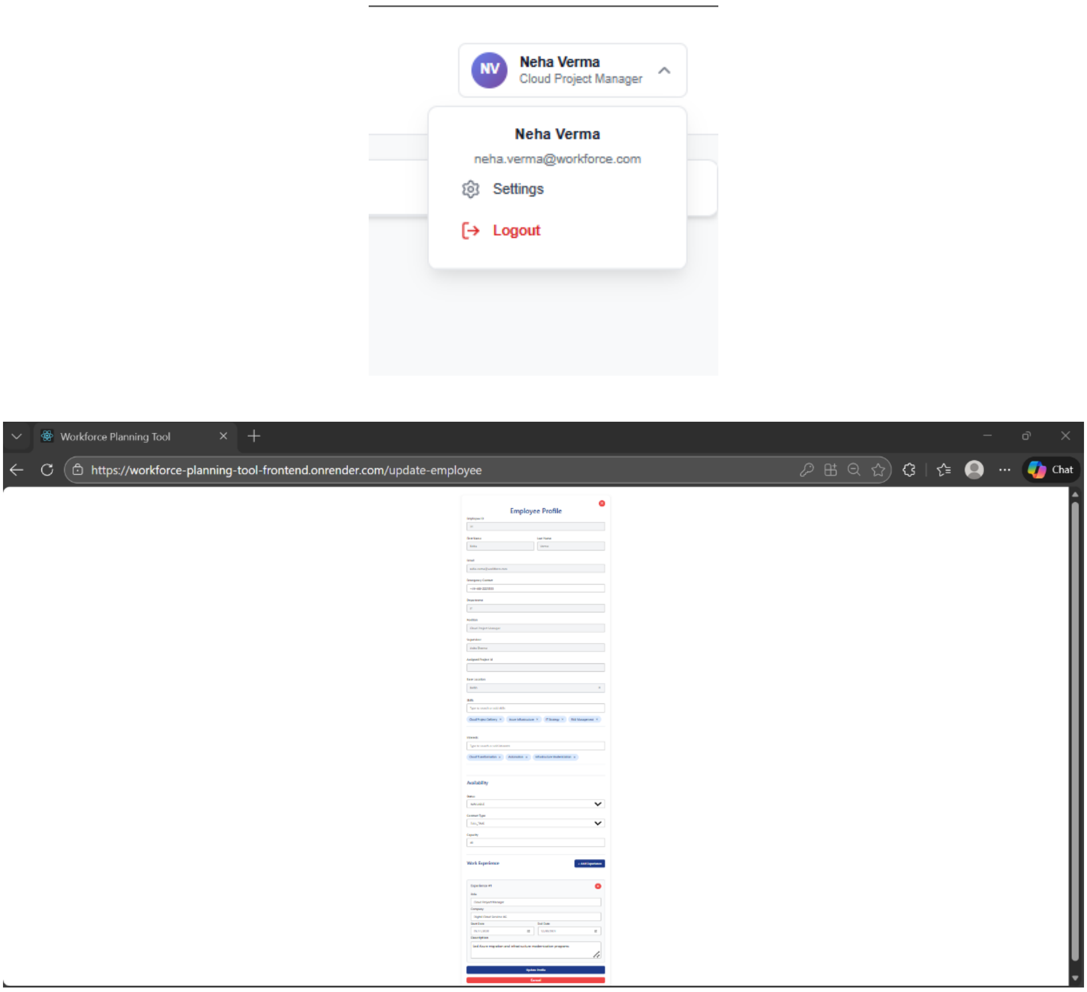


Figure 7: User Profile Menu

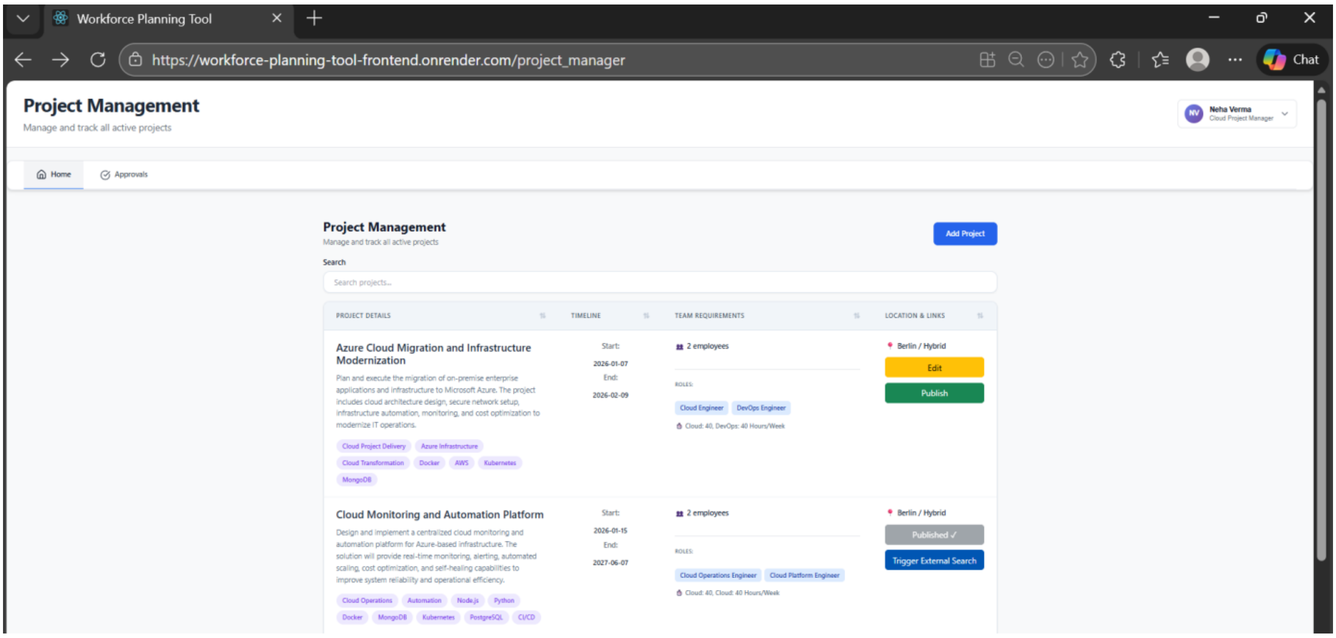
Project Management dashboard:

The Project Management Dashboard serves as the central workspace for users with the Project Manager role in the Workforce Management Tool. It provides a high level overview of project related activities and supports the core responsibilities of Project Managers. The dashboard is organized into key sections such as Home and Approvals, enabling Project Managers to manage projects and track workforce related requests in a structured manner.

Home Section:

The Home section provides a comprehensive overview of all projects managed by the Project Manager. It displays a structured list of projects along with essential information such as project title, description, timeline, team requirements, locations, and current publication status. This section enables Project Managers to quickly understand the state of projects and supports informed decision making related to workforce planning. A search functionality is provided at the top of the Home section, allowing Project Managers to filter projects by name or keywords. This feature improves usability, especially when managing multiple projects, by enabling fast access to specific project records. Each project entry clearly shows the number of required employees, defined roles, associated competencies, and capacity in hours per week, which directly aligns with the requirement to define staffing needs and workload. The Home section also includes actionable controls such as Add Project, Edit, Publish and Trigger External Search.

The Add Project function allows Project Managers to initiate new project requests, while the Edit option enables modification of existing project details. The Publish button is used to make projects visible for further approval and staffing workflows. The publication status indicator provides whether a project is still in preparation or has been finalized for workforce allocation. Once a project is published from this screen, it becomes visible to other roles in the system, marking the transition from internal project preparation to the approval and staffing phase. The Trigger External Search functionality allows the Project Manager to initiate an external workforce search when internal employee availability is insufficient. By enabling this action, the system supports and fulfills the requirement to extend staffing beyond internal resources when necessary thus successfully establishes interface to other project teams.



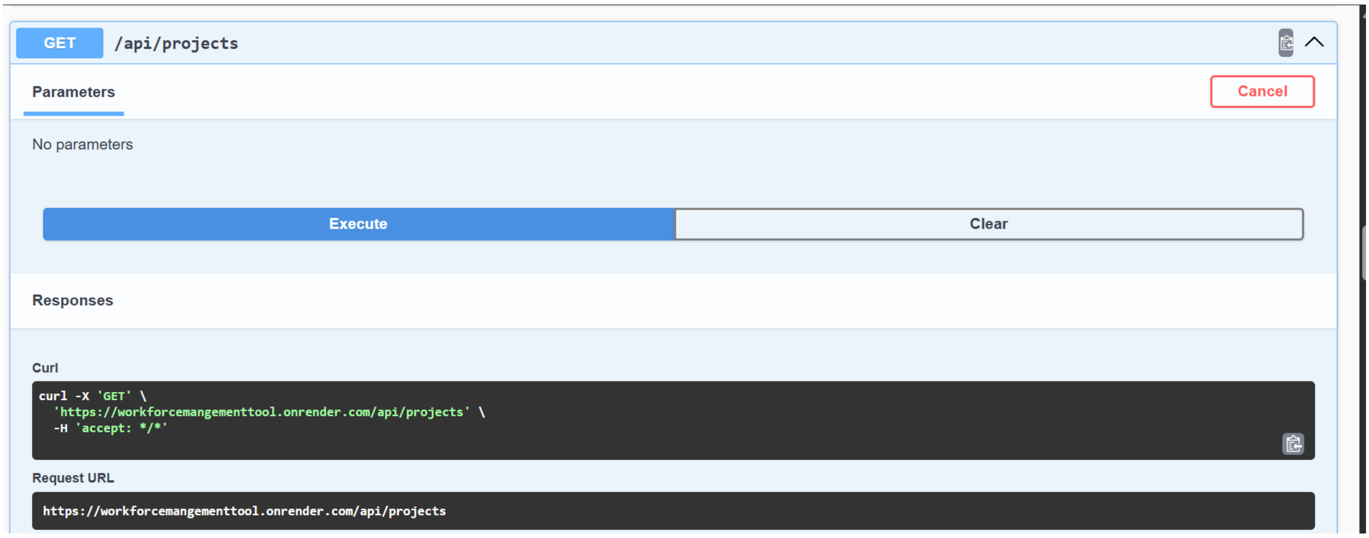
Project Management

Manage and track all active projects

Project Management

Manage and track all active projects

PROJECT DETAILS	TIMELINE	TEAM REQUIREMENTS	LOCATION & LINKS
Azure Cloud Migration and Infrastructure Modernization Plan and execute the migration of on-premise enterprise applications and infrastructure to Microsoft Azure. The project includes cloud architecture design, secure network setup, infrastructure automation, monitoring, and cost optimization to modernize IT operations. Cloud Project Delivery · Azure Infrastructure · Cloud Transformation · Docker · AWS · Kubernetes · MongoDB	Start: 2026-01-07 End: 2026-02-09	2 employees ROLES: Cloud Engineer · DevOps Engineer Cloud: 40, DevOps: 40 Hours/Week	Berlin / Hybrid
Cloud Monitoring and Automation Platform Design and implement a centralized cloud monitoring and automation platform for Azure-based infrastructure. The solution will provide real-time monitoring, alerting, automated scaling, cost optimization, and self-healing capabilities to improve system reliability and operational efficiency. Cloud Operations · Automation · Node.js · Python · Docker · MongoDB · Kubernetes · PostgreSQL · CI/CD	Start: 2026-01-15 End: 2027-06-07	2 employees ROLES: Cloud Operations Engineer · Cloud Platform Engineer Cloud: 40, Cloud: 40 Hours/Week	Berlin / Hybrid



/api/projects

Parameters

No parameters

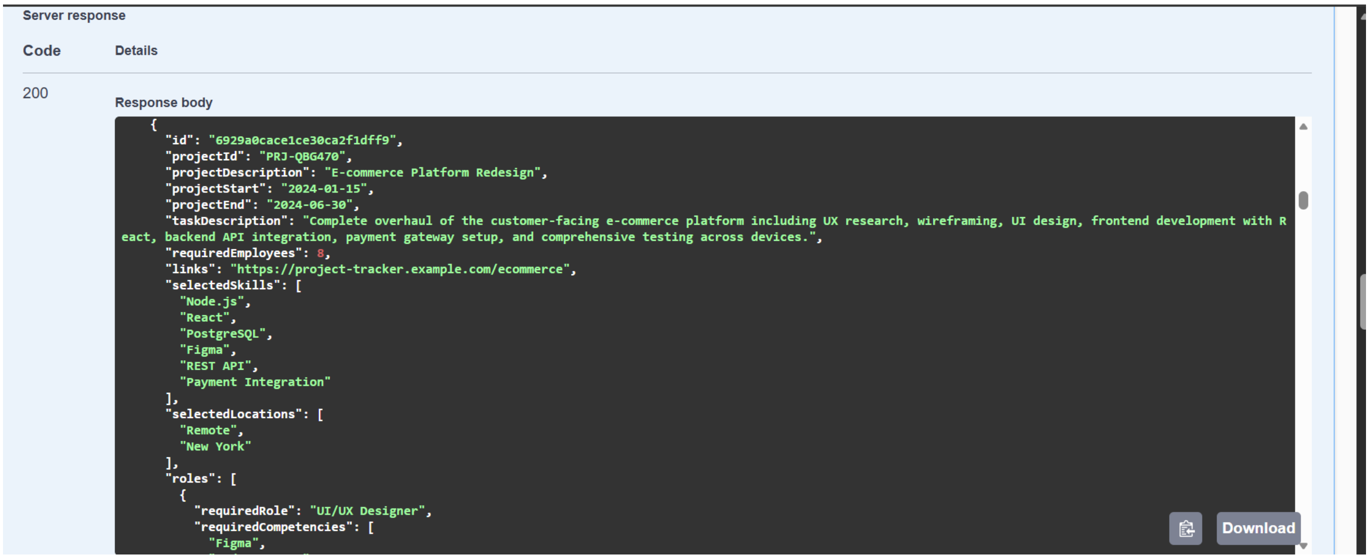
Responses

Curl

```
curl -X 'GET' \
  'https://workforcemangementtool.onrender.com/api/projects' \
  -H 'accept: */*'
```

Request URL

```
https://workforcemangementtool.onrender.com/api/projects
```



Server response

Code Details

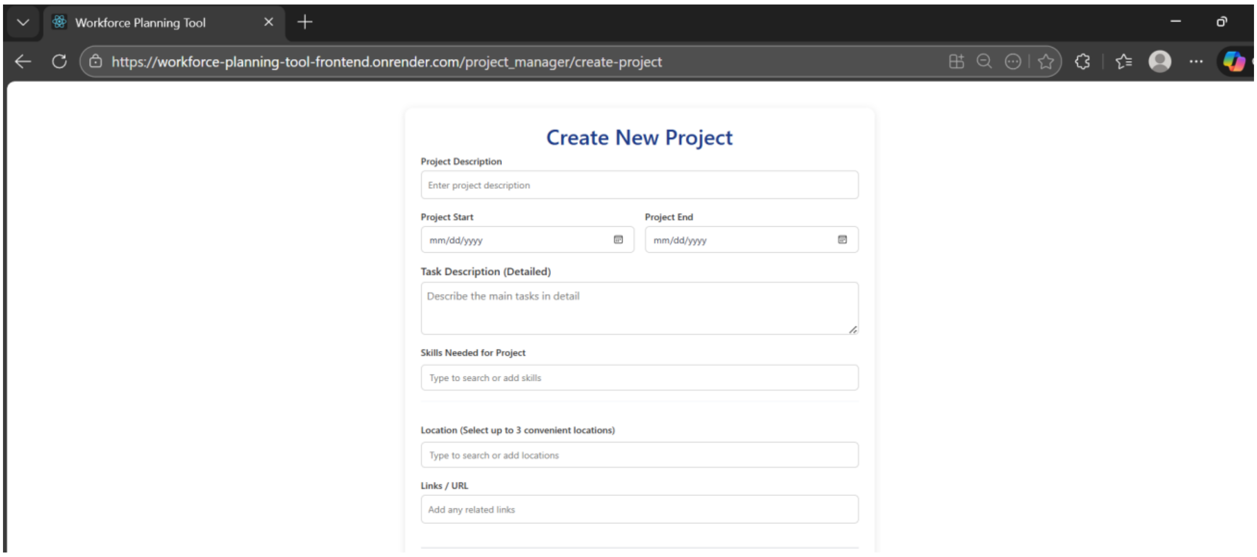
200 Response body

```
    {
      "id": "6929a0cace1ce30ca2f1dff9",
      "projectId": "PRJ-QBG470",
      "projectDescription": "E-commerce Platform Redesign",
      "projectStart": "2024-01-15",
      "projectEnd": "2024-06-30",
      "taskDescription": "Complete overhaul of the customer-facing e-commerce platform including UX research, wireframing, UI design, frontend development with R
eact, backend API integration, payment gateway setup, and comprehensive testing across devices.",
      "requiredEmployees": 8,
      "links": "https://project-tracker.example.com/ecommerce",
      "selectedSkills": [
        "Node.js",
        "React",
        "PostgreSQL",
        "Figma",
        "REST API",
        "Payment Integration"
      ],
      "selectedLocations": [
        "Remote",
        "New York"
      ],
      "roles": [
        {
          "requiredRole": "UI/UX Designer",
          "requiredCompetencies": [
            "Figma",
```

Figure 8: Home Section

Add Project:

The Create Project (Figure 9) view allows Project Managers to formally initiate a new project and define its staffing requirements. This view captures all required project metadata, including project description, start and end dates, and a detailed task description. These fields ensure that the scope, duration, and objectives of the project are clearly documented, fulfilling the requirement to define project context and timelines. In addition, the Create Project view enables Project Managers to specify skills needed for the project, preferred work locations, and relevant links or URLs. This information supports skill based matching and provides contextual references for stakeholders involved in the project. The interface allows selection or entry of multiple skills and locations, ensuring flexibility in workforce planning. A key component of this view is the Project Roles section, where Project Managers can define one or more roles required for the project. For each role, the required role name, number of employees, required competencies, and weekly capacity are specified. The ability to add multiple roles ensures that complex projects with diverse staffing needs can be accurately represented. Once all details are provided, the Create Project action submits the project for further processing in the system.



Workforce Planning Tool
https://workforce-planning-tool-frontend.onrender.com/project_manager/create-project

Create New Project

Project Description

Enter project description

Project Start: mm/dd/yyyy

Project End: mm/dd/yyyy

Task Description (Detailed)

Describe the main tasks in detail

Skills Needed for Project

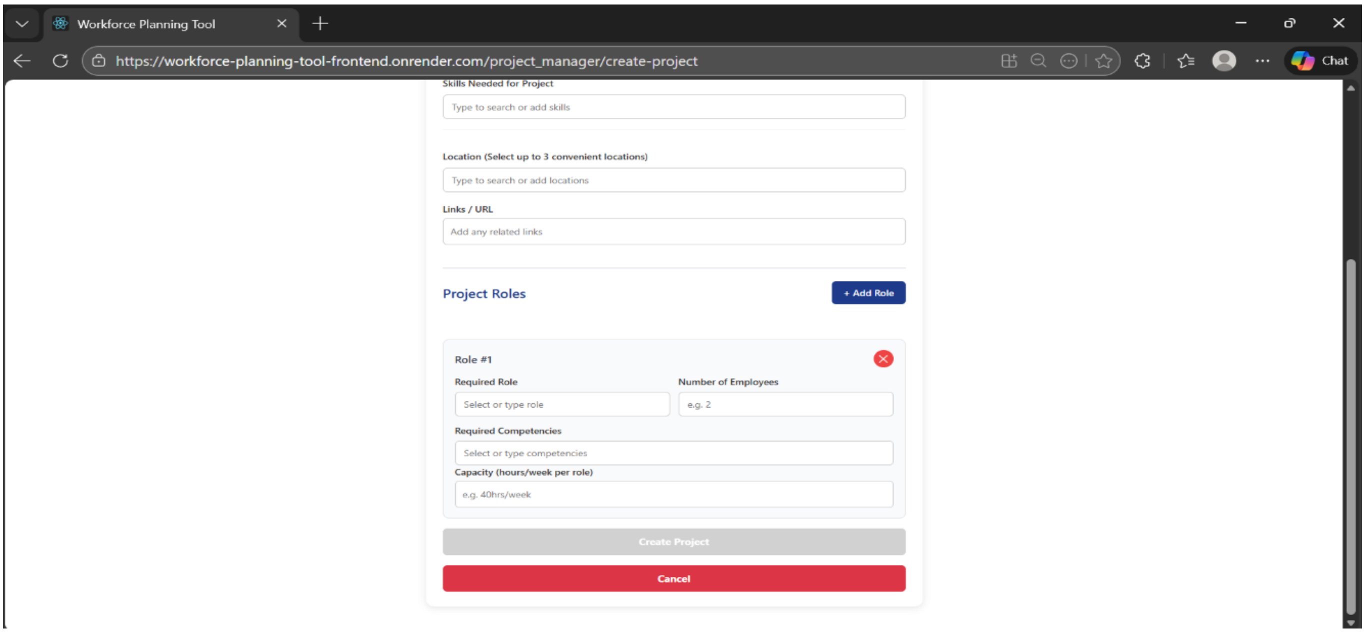
Type to search or add skills

Location (Select up to 3 convenient locations)

Type to search or add locations

Links / URL

Add any related links


Workforce Planning Tool
https://workforce-planning-tool-frontend.onrender.com/project_manager/create-project
Chat

Skills Needed for Project

Type to search or add skills

Location (Select up to 3 convenient locations)

Type to search or add locations

Links / URL

Add any related links

Project Roles

Role #1

Required Role: Select or type role

Number of Employees: e.g. 2

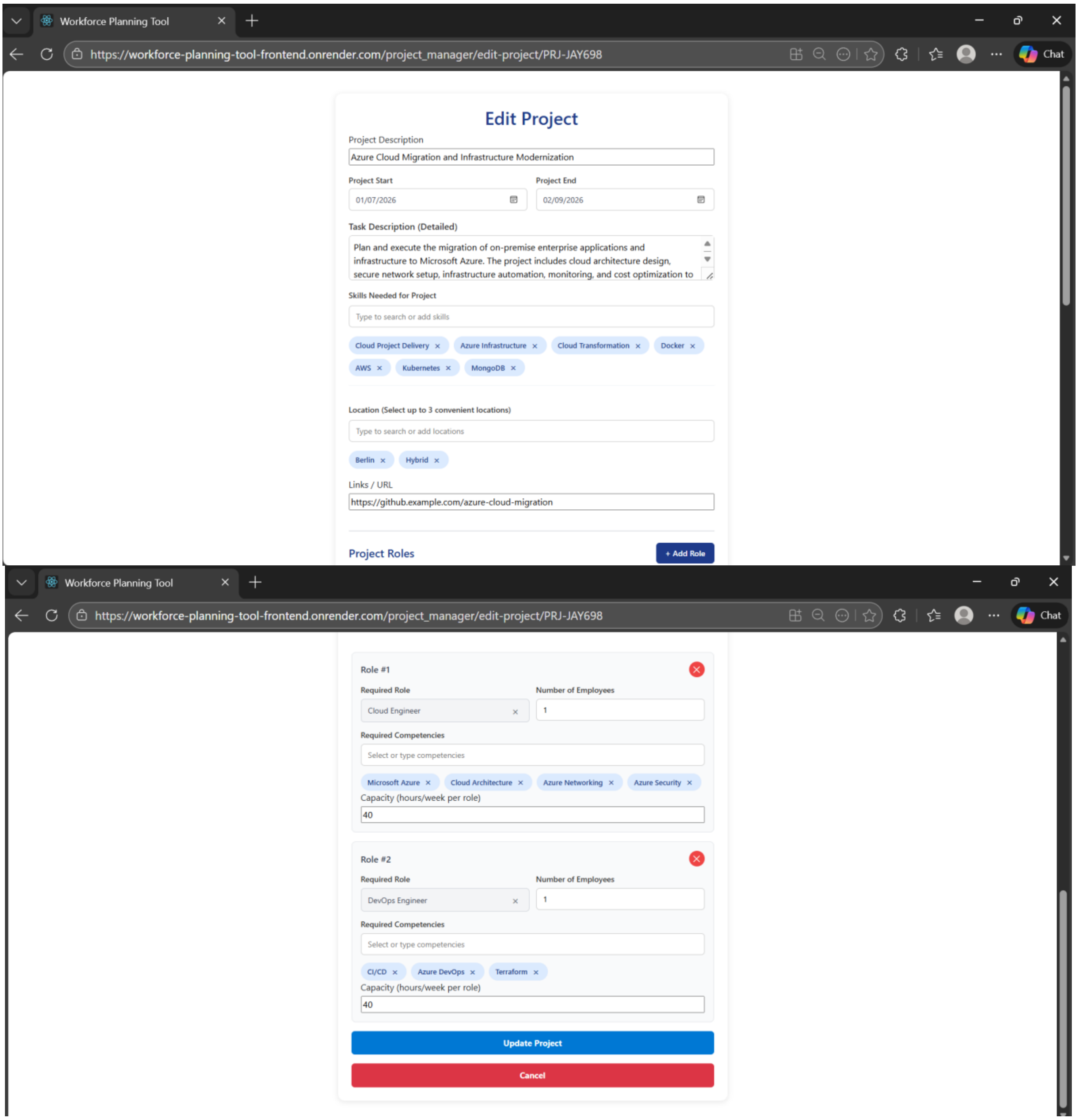
Required Competencies: Select or type competencies

Capacity (hours/week per role): e.g. 40hrs/week

Figure 9: Add Project

Edit Project View:

The Edit Project view allows Project Managers to update and refine existing project details after initial creation. This view displays previously entered project information, including description, timeline, task details, skills, locations, and associated links, enabling controlled modification as project requirements evolve. This functionality supports adaptability, allowing changes without recreating projects from scratch. This view also enables modification of project roles and staffing requirements. Project Managers can update role definitions, adjust the number of required employees, modify competencies, and change capacity values based on updated project needs. Roles can be added or removed dynamically, ensuring that workforce planning remains accurate and up to date. After making changes, the Project Manager can save updates using the Update Project action or cancel modifications if needed. Once updates are saved, the revised project details are reflected in the Home section. When the Project Manager publishes the updated project, it becomes available for approval and further workforce planning, continuing the workflow toward Department Head review and employee assignment.


Workforce Planning Tool
https://workforce-planning-tool-frontend.onrender.com/project_manager/edit-project/PRJ-JAY698
Chat

Edit Project

Project Description: Azure Cloud Migration and Infrastructure Modernization

Project Start: 01/07/2026

Project End: 02/09/2026

Task Description (Detailed): Plan and execute the migration of on-premise enterprise applications and infrastructure to Microsoft Azure. The project includes cloud architecture design, secure network setup, infrastructure automation, monitoring, and cost optimization to

Skills Needed for Project: Type to search or add skills

Cloud Project Delivery × Azure Infrastructure × Cloud Transformation × Docker × AWS × Kubernetes × MongoDB ×

Location (Select up to 3 convenient locations): Type to search or add locations

Berlin × Hybrid ×

Links / URL: https://github.example.com/azure-cloud-migration

Project Roles

+ Add Role

Role #1

Required Role: Cloud Engineer ×

Number of Employees: 1

Required Competencies: Select or type competencies

Microsoft Azure × Cloud Architecture × Azure Networking × Azure Security ×

Capacity (hours/week per role): 40

Role #2

Required Role: DevOps Engineer ×

Number of Employees: 1

Required Competencies: Select or type competencies

CI/CD × Azure DevOps × Terraform ×

Capacity (hours/week per role): 40

Update Project

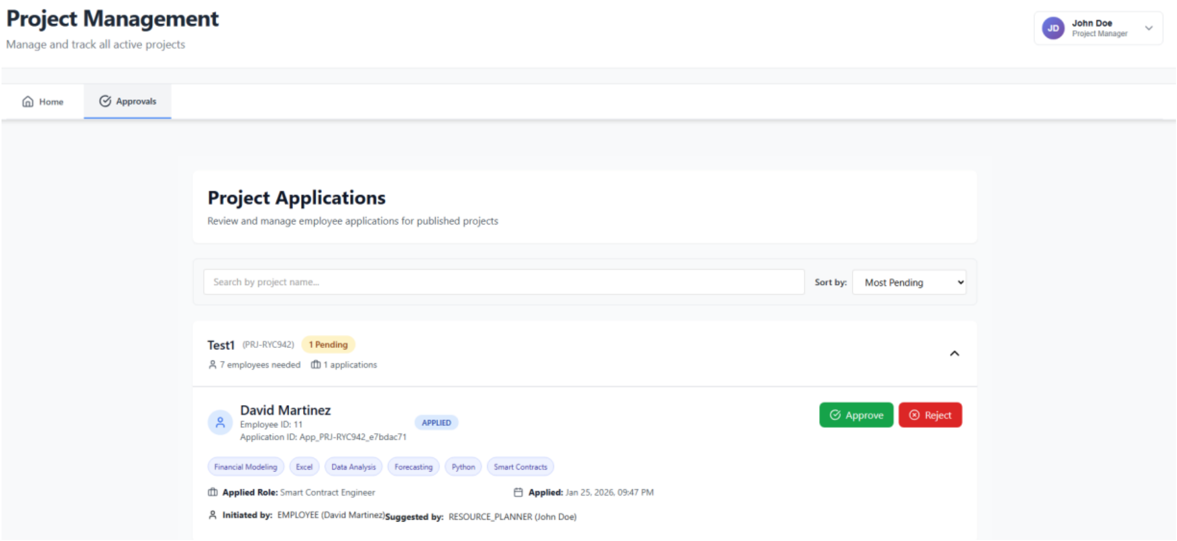
Cancel

Figure 10: Edit Project

Approvals Section:

The Approvals section in Figure 11, allows the Project Manager to review and manage employee applications for projects that have already been published. This section lists published projects along with key indicators such as the number of employees required and the number of applications received. With this information, the system enables the Project Manager to monitor interest and engagement from employees who have applied for roles within a project and the Project Manager evaluates potential candidates proposed through the system. A search bar allows filtering applications by project name, while sorting options will help the Project Manager prioritize review activities efficiently when multiple applications are awaiting action. When employee applications are available, the Project Manager can expand a project entry to review applicant details and assess suitability based on project needs. The

Project Manager can either Approve or Reject each application. Upon approval, the employee is formally assigned to the project, and the project's staffing requirement is updated accordingly in the system. Rejection immediately removes the application, allowing alternative candidates to be considered.



Project Management
Manage and track all active projects

Project Applications
Review and manage employee applications for published projects

Search by project name...
Sort by: Most Pending

Test1 (PRJ-RYC942) 1 Pending
7 employees needed 1 applications

David Martinez
Employee ID: 11
Application ID: App_PRJ-RYC942_e7bdac71
APPLIED

Financial Modeling Excel Data Analysis Forecasting Python Smart Contracts

Applied Role: Smart Contract Engineer
Applied: Jan 25, 2026, 09:47 PM
Initiated by: EMPLOYEE (David Martinez)Suggested by: RESOURCE_PLANNER (John Doe)

Figure 11: Approvals Section

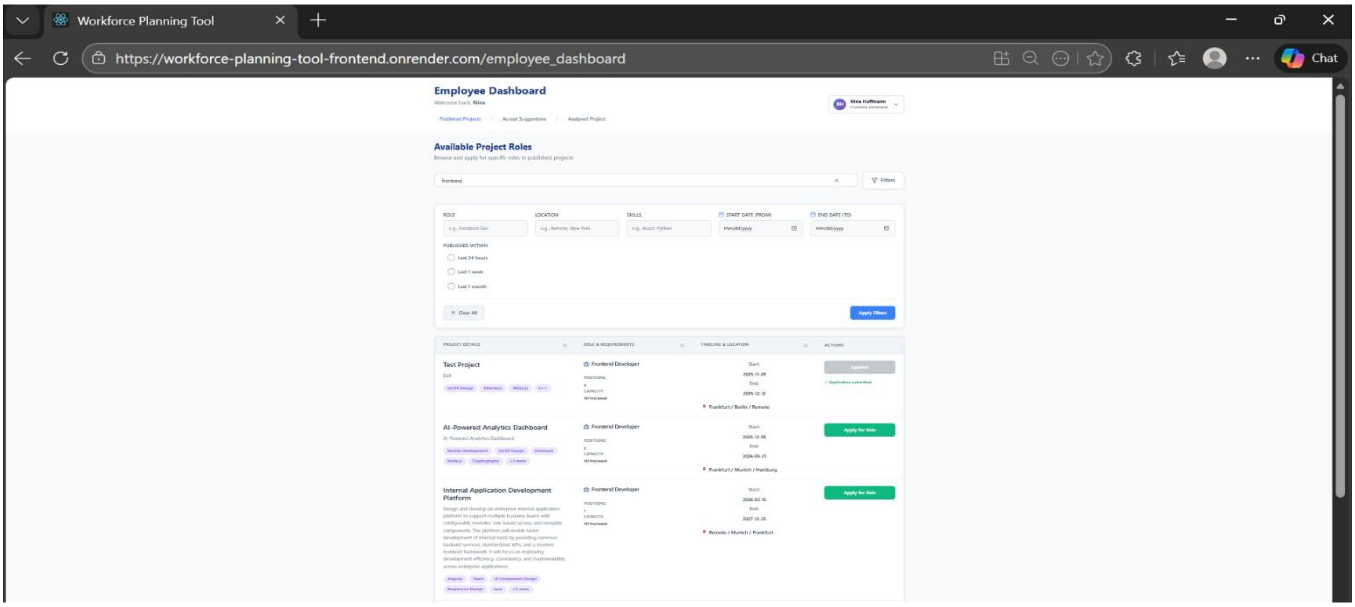
Employee dashboard:

The Employee Dashboard serves as the primary interface for employees to interact with clear visibility into available project opportunities and their confirmed project assignments. This dashboard ensures that employees can actively engage in the allocation process regarding their current and potential involvement. The Employee Dashboard is organized into Published Projects, Accept Suggestions, and Assigned Project. Each section represents a distinct phase of employee interaction within the workforce planning workflow. This allows employees to browse open projects, respond to recommendations, and track their assigned work in a clear manner, supporting effective collaboration between employees, Project Managers, and other organizational roles.

Published Projects section:

The Published Projects section allows employees to browse all active and approved projects. This section allow employees to discover suitable project roles based on their skills, availability, and location preferences. Only projects that have been published by the Project Manager appear here, ensuring employees view only validated and approved staffing opportunities. This section provides search and advanced filtering options to refine project listings. Filters include role type, location, required skills, project start and end dates, and publication time range (such as projects published within the last 24 hours, week, or month). Each project displays information including the project name, description, required role, number of positions, weekly capacity, project timeline, and work location. Required skills are clearly highlighted using tags and it shows skill based matching between employees and project needs.

In this section (Figure 12), employees can directly apply for a role using the “Apply for Role” action. Once an application is submitted, the status changes to indicate successful submission and the application is then routed to the Project Manager’s Approvals section, where it enters the formal review and decision. This ensures a seamless, traceable flow from employee interest to managerial approval within the workforce planning process.


Workforce Planning Tool
https://workforce-planning-tool-frontend.onrender.com/employee_dashboard
Chat

Employee Dashboard

Welcome back, Nina

Nina Hoffmann
Frontend Developer

Published Projects / Accept Suggestions / Assigned Project

Available Project Roles

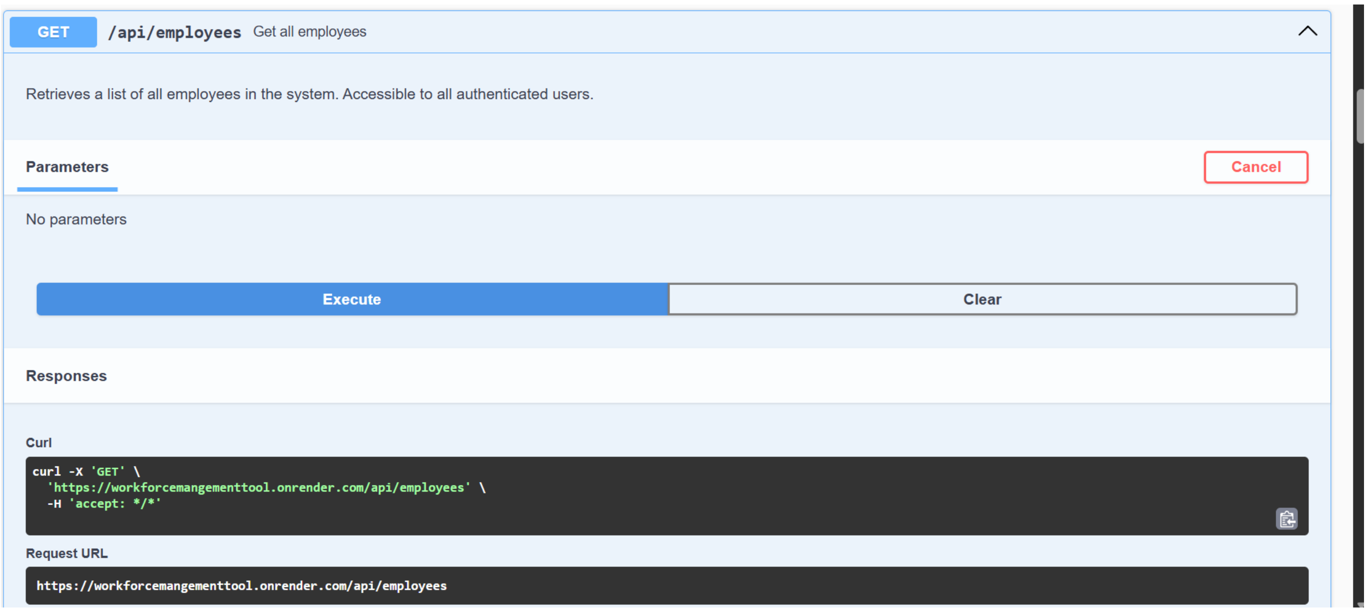
Browse and apply for specific roles in published projects

frontend

ROLE: e.g., Frontend Dev
LOCATION: e.g., Remote, New York
SKILLS: e.g., React, Python
START DATE (FROM): mm/dd/yyyy
END DATE (TO): mm/dd/yyyy

PUBLISHED WITHIN
○ Last 24 hours
○ Last 1 week
○ Last 1 month

PROJECT DETAILS	ROLE & REQUIREMENTS	TIMELINE & LOCATION	ACTIONS
Test Project Edit UI/UX Design, Ethereum, Web3.js, C++	Frontend Developer POSITIONS: 4 CAPACITY: 40 hrs/week	Start: 2025-11-29 End: 2025-12-12 Frankfurt / Berlin / Remote	Applied ✓ Application submitted
AI-Powered Analytics Dashboard AI-Powered Analytics Dashboard Mobile Development, UI/UX Design, Ethereum, Node.js, Cryptography, +3 more	Frontend Developer POSITIONS: 2 CAPACITY: 40 hrs/week	Start: 2025-12-05 End: 2026-05-21 Frankfurt / Munich / Hamburg	Apply for Role
Internal Application Development Platform Design and develop an enterprise internal application platform to support multiple business teams with configurable modules, role based access, and reusable components. The platform will enable faster development of internal tools by providing common backend services, standardized APIs, and a modern frontend framework. It will focus on improving development efficiency, consistency, and maintainability across enterprise applications. Angular, React, UI Component Design, Responsive Design, Java, +5 more	Frontend Developer POSITIONS: 1 CAPACITY: 40 hrs/week	Start: 2026-02-15 End: 2027-12-26 Remote / Munich / Frankfurt	Apply for Role



/api/employees Get all employees

Retrieves a list of all employees in the system. Accessible to all authenticated users.

Parameters

No parameters

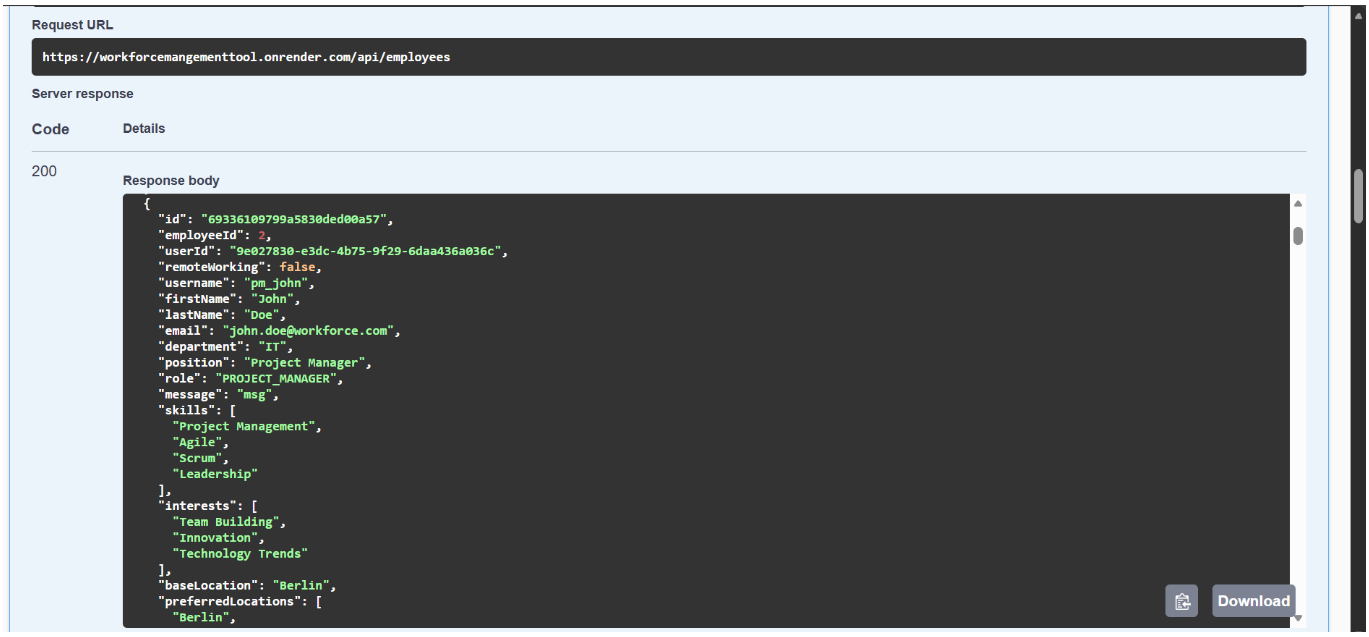
Responses

Curl

```
curl -X 'GET' \
  'https://workforcemangementtool.onrender.com/api/employees' \
  -H 'accept: */*'
```

Request URL

```
https://workforcemangementtool.onrender.com/api/employees
```



Request URL

```
https://workforcemangementtool.onrender.com/api/employees
```

Server response

Code Details

200 Response body

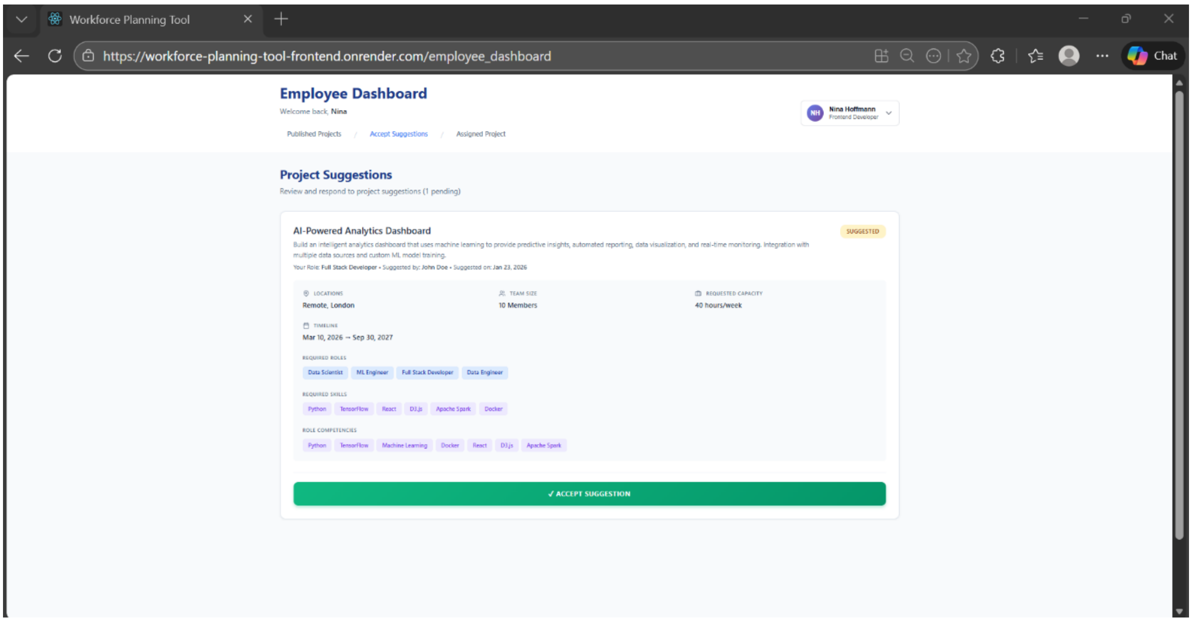
```
{
  "id": "69336109799a5830ded00a57",
  "employeeId": 2,
  "userId": "9e027830-e3dc-4b75-9f29-6daa436a036c",
  "remoteWorking": false,
  "username": "pm_john",
  "firstName": "John",
  "lastName": "Doe",
  "email": "john.doe@workforce.com",
  "department": "IT",
  "position": "Project Manager",
  "role": "PROJECT_MANAGER",
  "message": "msg",
  "skills": [
    "Project Management",
    "Agile",
    "Scrum",
    "Leadership"
  ],
  "interests": [
    "Team Building",
    "Innovation",
    "Technology Trends"
  ],
  "baseLocation": "Berlin",
  "preferredLocations": [
    "Berlin",
```

Figure 12: Published Projects section

Accept Suggestions Section:

This section in which Figure 13, provides project opportunities that have been actively suggested to the employee by the Resource Planner based on skill matching, availability, and workload balancing. Each suggestion provides detailed visibility into why the project was recommended. It displays the project name, a clear description of the business objective, the employee's suggested role, and the person who initiated the suggestion. Additional contextual information such as project location, total team size, required weekly capacity, and project timeline ensures the employee can evaluate the commitment before responding.

The section also highlights required roles, skills, and role competencies, allowing employees to verify with their technical expertise and help employees to make decisions by directly using the "Accept Suggestion" action. Once accepted, the employee's confirmation is recorded in the system and forwarded to the Project Manager for final approval. If not accepted, the suggestion remains pending or can be rebalanced by the Resource Planner.



Workforce Planning Tool
https://workforce-planning-tool-frontend.onrender.com/employee_dashboard
Chat
NH
Nina Hoffmann
Frontend Developer
Published Projects / Accept Suggestions / Assigned Project
SUGGESTED
✓ ACCEPT SUGGESTION

Employee Dashboard

Welcome back, Nina

Project Suggestions

Review and respond to project suggestions (1 pending)

AI-Powered Analytics Dashboard

Build an intelligent analytics dashboard that uses machine learning to provide predictive insights, automated reporting, data visualization, and real-time monitoring. Integration with multiple data sources and custom ML model training.

Your Role: Full Stack Developer • Suggested by: John Doe • Suggested on: Jan 23, 2026

LOCATIONS: Remote, London

TEAM SIZE: 10 Members

REQUESTED CAPACITY: 40 hours/week

TIMELINE: Mar 10, 2026 → Sep 30, 2027

REQUIRED ROLES: Data Scientist, ML Engineer, Full Stack Developer, Data Engineer

REQUIRED SKILLS: Python, TensorFlow, React, D3.js, Apache Spark, Docker

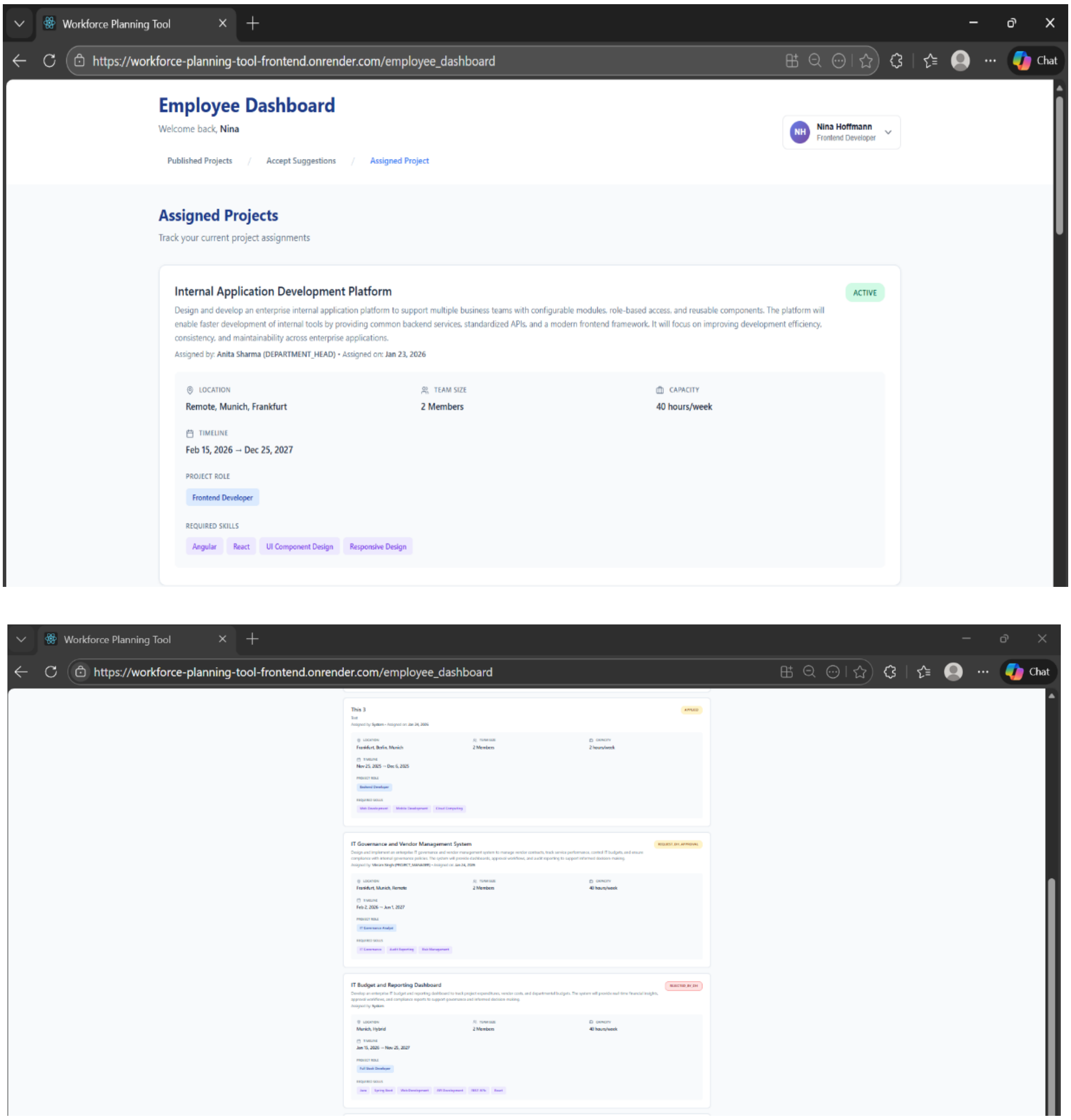
ROLE COMPETENCIES: Python, TensorFlow, Machine Learning, Docker, React, D3.js, Apache Spark

Figure 13: Accept Suggestions Section

Assigned Projects Section:

The Assigned Projects section provides employees with a view of all projects to which they are currently assigned. This section helps employees clearly understand their current workload, role responsibilities, and assignment status within the organization. Each project in this section displays assignment details, including the project name, detailed description, assignment status, location, timeline, team size, and weekly capacity requirements. The system also explicitly shows who assigned the project (for example, a Department Head or system). The section differentiates assignment states using status indicators such as active, applied, and rejected_by_dh.

1. ACTIVE indicates that the employee has been fully approved and is currently working on the project.
2. APPLIED shows that the employee has applied for a role and the request is still under review.
3. REQUEST_DH_APPROVAL indicates that the employee's assignment has been forwarded by the Project Manager and is currently awaiting approval from the Department Head.
4. REJECTED_BY_DH indicates that the assignment was not approved at the department head level.


Workforce Planning Tool
https://workforce-planning-tool-frontend.onrender.com/employee_dashboard
Chat

Employee Dashboard

Welcome back, **Nina**

NH Nina Hoffmann
Frontend Developer

Published Projects / Accept Suggestions / Assigned Project

Assigned Projects

Track your current project assignments

Internal Application Development Platform ACTIVE

Design and develop an enterprise internal application platform to support multiple business teams with configurable modules, role-based access, and reusable components. The platform will enable faster development of internal tools by providing common backend services, standardized APIs, and a modern frontend framework. It will focus on improving development efficiency, consistency, and maintainability across enterprise applications.

Assigned by: Anita Sharma (DEPARTMENT_HEAD) • Assigned on: Jan 23, 2026

LOCATION: Remote, Munich, Frankfurt
TEAM SIZE: 2 Members
CAPACITY: 40 hours/week
TIMELINE: Feb 15, 2026 → Dec 25, 2027
PROJECT ROLE: Frontend Developer
REQUIRED SKILLS: Angular, React, UI Component Design, Responsive Design

This 3 APPLIED

Test

Assigned by: System • Assigned on: Jan 24, 2026

LOCATION: Frankfurt, Berlin, Munich
TEAM SIZE: 2 Members
CAPACITY: 2 hours/week
TIMELINE: Nov 25, 2025 → Dec 6, 2025
PROJECT ROLE: Backend Developer
REQUIRED SKILLS: Web Development, Mobile Development, Cloud Computing

IT Governance and Vendor Management System REQUEST_DH_APPROVAL

Design and implement an enterprise IT governance and vendor management system to manage vendor contracts, track service performance, control IT budgets, and ensure compliance with internal governance policies. The system will provide dashboards, approval workflows, and audit reporting to support informed decision making.

Assigned by: Vikram Singh (PROJECT_MANAGER) • Assigned on: Jan 24, 2026

LOCATION: Frankfurt, Munich, Remote
TEAM SIZE: 2 Members
CAPACITY: 40 hours/week
TIMELINE: Feb 2, 2026 → Jun 1, 2027
PROJECT ROLE: IT Governance Analyst
REQUIRED SKILLS: IT Governance, Audit Reporting, Risk Management

IT Budget and Reporting Dashboard REJECTED_BY_DH

Develop an enterprise IT budget and reporting dashboard to track project expenditures, vendor costs, and departmental budgets. The system will provide real-time financial insights, approval workflows, and compliance reports to support governance and informed decision making.

Assigned by: System

LOCATION: Munich, Hybrid
TEAM SIZE: 2 Members
CAPACITY: 40 hours/week
TIMELINE: Jan 15, 2026 → Nov 25, 2027
PROJECT ROLE: Full Stack Developer
REQUIRED SKILLS: Java, Spring Boot, Web Development, API Development, REST APIs, React

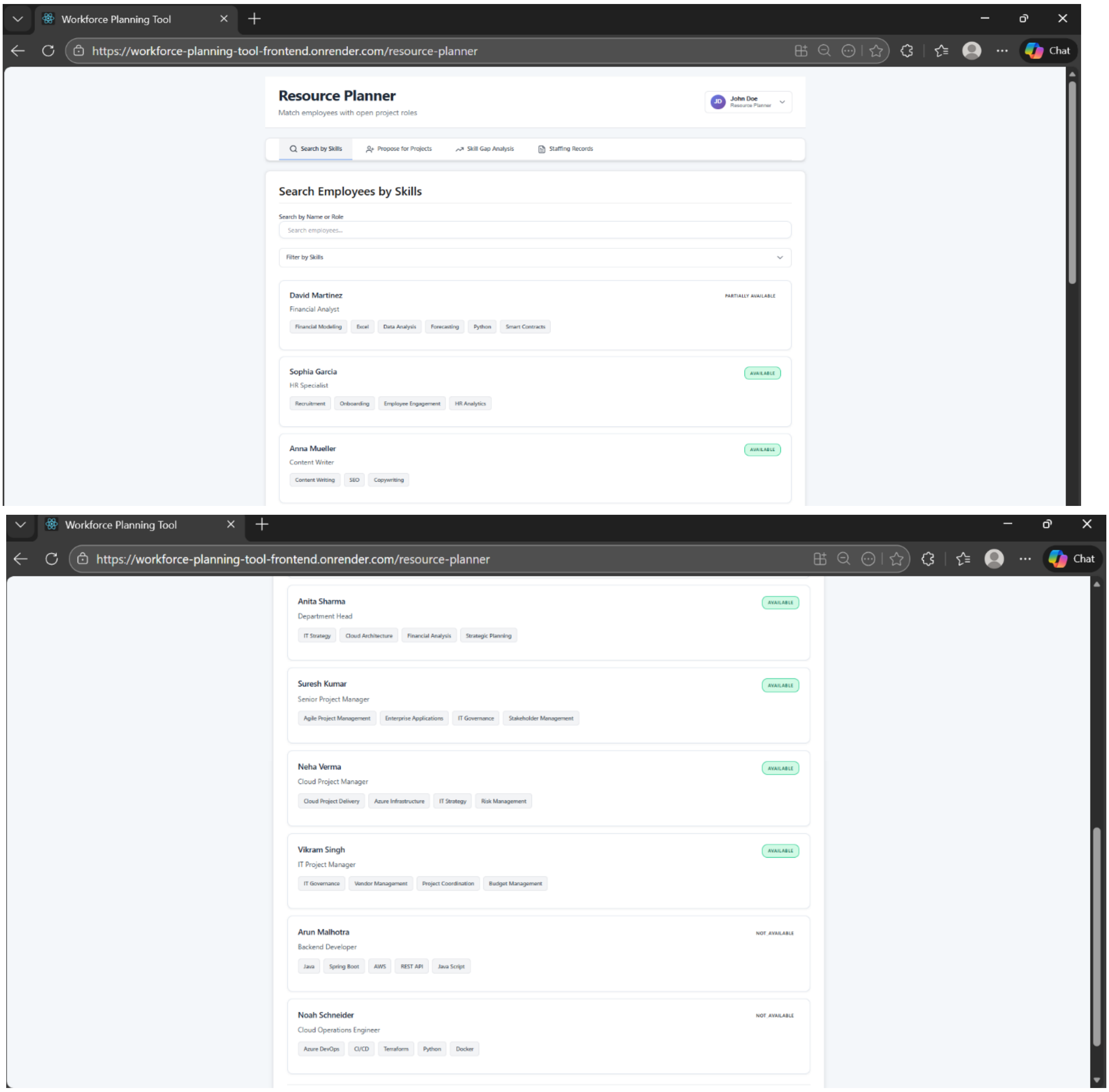
Figure 14: Assigned Projects Section

Resource Planner dashboard:

The Resource Planner Dashboard is designed to support the core function of matching available employees to open project roles in an efficient manner. The Resource Planner acts as an intermediary role between Project Managers, Employees, and Department Heads, ensuring that staffing decisions are based on skills, availability, and workload. The Resource Planner is responsible for viewing available employees, searching and filtering them by skills and competencies, and proposing suitable candidates for project roles defined by Project Managers. The dashboard enables the Resource Planner to perform a skill gap analysis, identifying where internal capabilities are sufficient and where external hiring may be required. This supports the system’s objective of optimizing internal resource utilization before triggering external searches. While the Resource Planner can recommend or suggest employees for projects, the final decision is routed through the Department Head approval process and also maintaining upto date information on employee availability and project staffing records.

Search by skills Section:

The Search by Skills section enables the Resource Planner to efficiently identify suitable employees for open project roles based on skills, roles, and availability, ensuring optimal resource allocation across projects. In this section, the Resource Planner can search employees using name or role based keywords and further refine results using skill based filters. Each employee displays information such as the employee’s name, current role, key competencies, and availability status (Available, Partially Available, or Not Available). This visibility helps the Resource Planner quickly assess whether an employee can be considered for a specific project. Employees marked as Available can be proposed immediately for new assignments, while Partially Available resources allow planners to balance workloads across overlapping projects. Employees marked as Not Available are excluded from active consideration.



Resource Planner

Match employees with open project roles

Search Employees by Skills

Search by Name or Role

Search employees...

Filter by Skills

David Martinez — PARTIALLY AVAILABLE
Financial Analyst
Financial Modeling | Excel | Data Analysis | Forecasting | Python | Smart Contracts

Sophia Garcia — AVAILABLE
HR Specialist
Recruitment | Onboarding | Employee Engagement | HR Analytics

Anna Mueller — AVAILABLE
Content Writer
Content Writing | SEO | Copywriting

Workforce Planning Tool
https://workforce-planning-tool-frontend.onrender.com/resource-planner
Chat

Anita Sharma — AVAILABLE
Department Head
IT Strategy | Cloud Architecture | Financial Analysis | Strategic Planning

Suresh Kumar — AVAILABLE
Senior Project Manager
Agile Project Management | Enterprise Applications | IT Governance | Stakeholder Management

Neha Verma — AVAILABLE
Cloud Project Manager
Cloud Project Delivery | Azure Infrastructure | IT Strategy | Risk Management

Vikram Singh — AVAILABLE
IT Project Manager
IT Governance | Vendor Management | Project Coordination | Budget Management

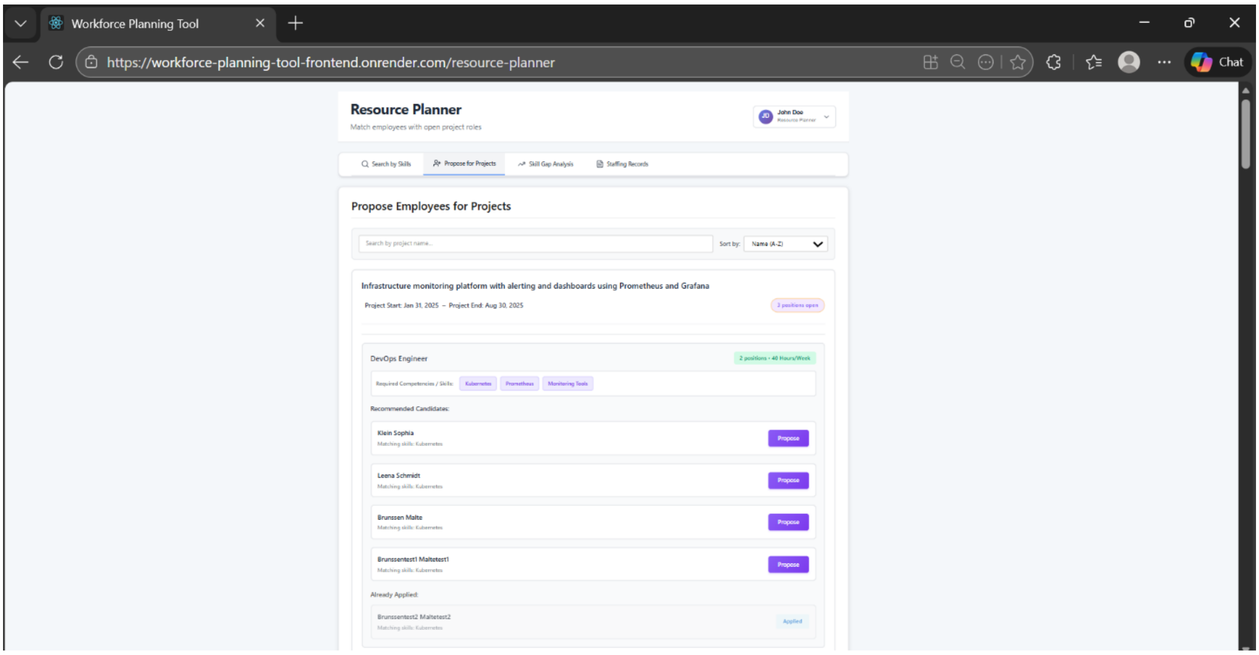
Arun Malhotra — NOT_AVAILABLE
Backend Developer
Java | Spring Boot | AWS | REST API | Java Script

Noah Schneider — NOT_AVAILABLE
Cloud Operations Engineer
Azure DevOps | CI/CD | Terraform | Python | Docker

Figure 15: Search by Skills Section

Propose for projects Section:

The Propose for Projects section enables the Resource Planner to actively assign suitable employees to open project roles in a structured manner. In this section, the Resource Planner can search for projects by name and sort them for easier navigation. Each project displays key information such as the project title, start and end dates, and the number of open positions. Within each project, roles are clearly listed along with required competencies allowing the Resource Planner to understand staffing needs at a glance. For every role, the system automatically suggests recommended candidates based on skill matching and availability. The Propose action allows the Resource Planner to formally suggest an employee for a specific project role. Once proposed, the employee's status is updated to Applied and the request moves forward for further approval.



Resource Planner

Match employees with open project roles

Propose Employees for Projects

Infrastructure monitoring platform with alerting and dashboards using Prometheus and Grafana

Project Start: Jan 31, 2025 – Project End: Aug 30, 2025

3 positions open

DevOps Engineer

2 positions • 40 Hours/Week

Required Competencies / Skills: Kubernetes Prometheus Monitoring Tools

Recommended Candidates:

Klein Sophia
Matching skills: Kubernetes

Leena Schmidt
Matching skills: Kubernetes

Brunssen Malte
Matching skills: Kubernetes

Brunssentest1 Maltetest1
Matching skills: Kubernetes

Already Applied:

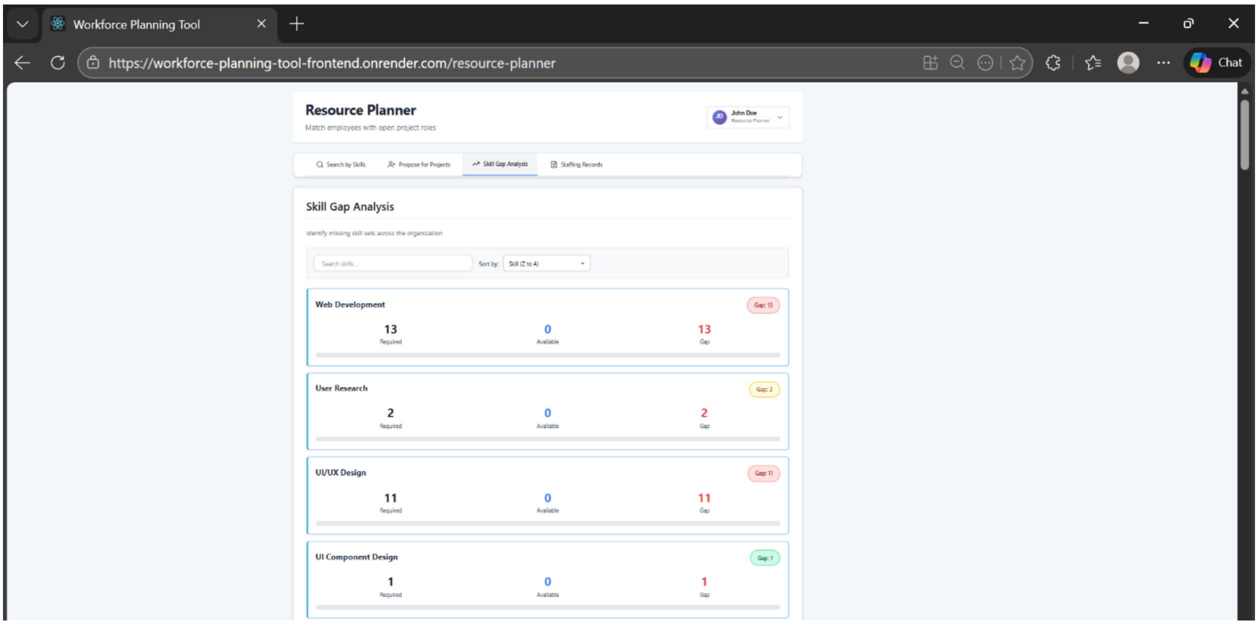
Brunssentest2 Maltetest2
Matching skills: Kubernetes

Figure 16: Propose for Projects Section

Skill Gap Analysis Section:

This section helps the Resource Planner identify mismatches between project skill demand and current employee skill availability across the organization. This feature supports by highlighting areas where talent shortages exist. In this section, the Resource Planner can search for specific skills and sort them (for example, from Z–A) to prioritize analysis. Each skill is displayed as an individual and showing three key metrics: Required, Available, and Gap.

- Required represents the total number of roles across active and upcoming projects that demand this skill.
- Available shows how many employees currently possess the skill and are available for assignment.
- Gap is the difference between required and available resources, clearly indicating shortages.


Workforce Planning Tool
https://workforce-planning-tool-frontend.onrender.com/resource-planner
Chat

Resource Planner

Match employees with open project roles

JD John Doe
Resource Planner

Search by Skills | Propose for Projects | Skill Gap Analysis | Staffing Records

Skill Gap Analysis

Identify missing skill sets across the organization

Search skills... Sort by: Skill (Z to A)

Web Development — Gap: 13
13 Required | 0 Available | 13 Gap

User Research — Gap: 2
2 Required | 0 Available | 2 Gap

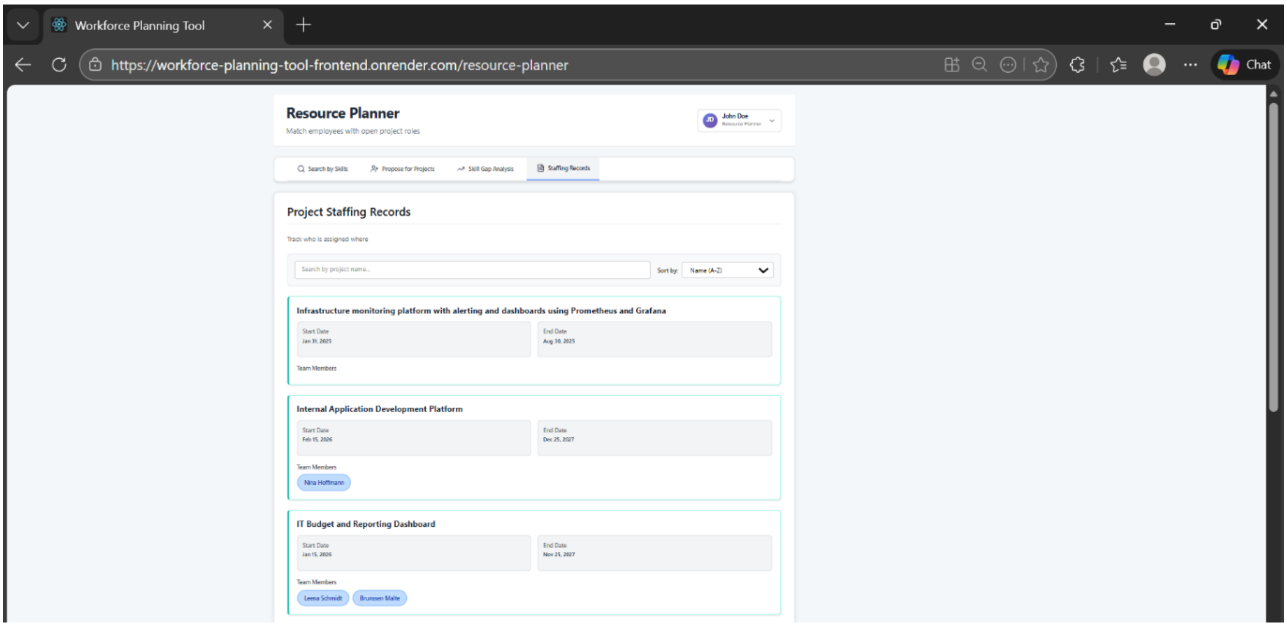
UI/UX Design — Gap: 11
11 Required | 0 Available | 11 Gap

UI Component Design — Gap: 1
1 Required | 0 Available | 1 Gap

Figure 17: Skill Gap Analysis Section

Staffing Records Section:

The Staffing Records section provides who is assigned to which project to the the Resource Planner, to track allocation across the organization. This section acts as the final confirmation layer after proposals have been approved and employees have been officially assigned to projects. The Resource Planner can search projects by name and sort records. Each project card displays staffing details, including project name, start date, and end date and the team Members field clearly lists the employees currently assigned to that project. Projects without listed members indicate roles that are either still open or pending assignment. Overall this section reflects the end state of the staffing lifecycle that the Projects are created and published by the Project Manager and the Employees apply or are proposed by the Resource Planner and Project Managers approve assignments and the approved assignments are recorded and displayed here.


Workforce Planning Tool
https://workforce-planning-tool-frontend.onrender.com/resource-planner
Chat

Resource Planner

Match employees with open project roles

JD John Doe
Resource Planner

Search by Skills | Propose for Projects | Skill Gap Analysis | Staffing Records

Project Staffing Records

Track who is assigned where

Search by project name... Sort by: Name (A-Z)

Infrastructure monitoring platform with alerting and dashboards using Prometheus and Grafana
Start Date: Jan 31, 2025 | End Date: Aug 30, 2025
Team Members

Internal Application Development Platform
Start Date: Feb 15, 2026 | End Date: Dec 25, 2027
Team Members
Nina Hoffmann

IT Budget and Reporting Dashboard
Start Date: Jan 15, 2026 | End Date: Nov 25, 2027
Team Members
Leena Schmidt | Brunssen Malte

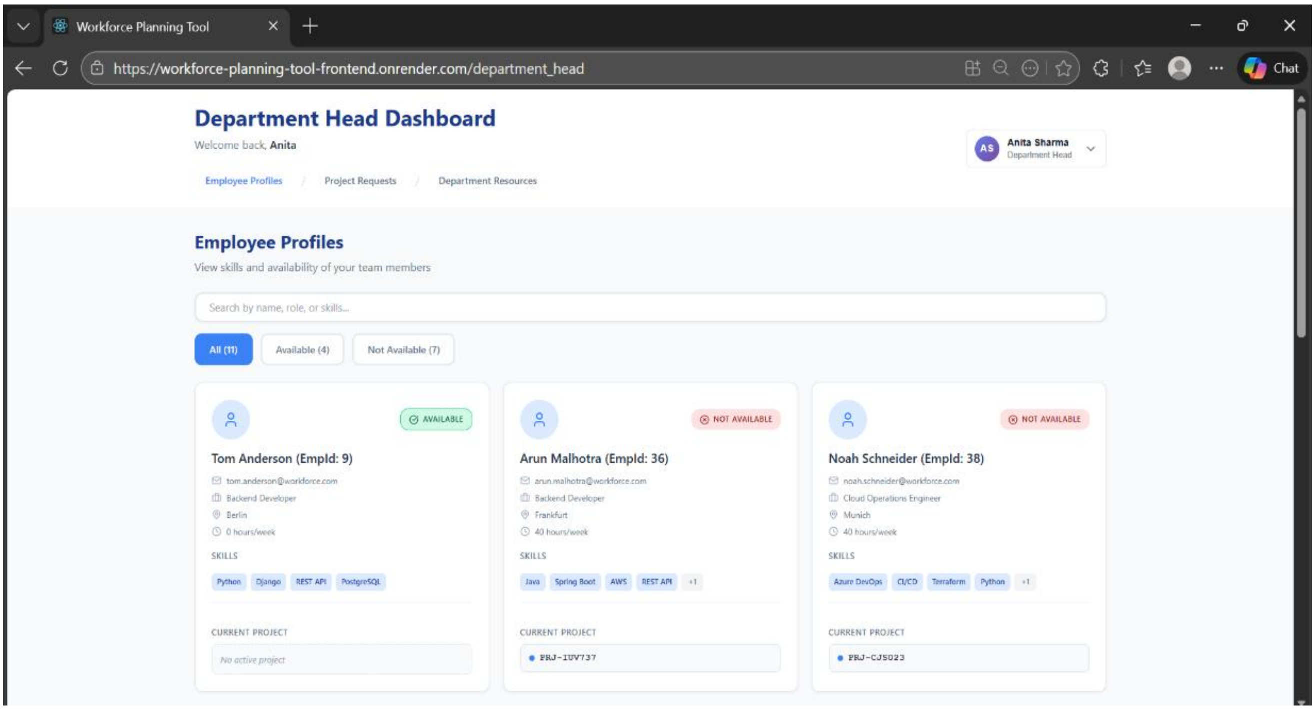
Figure 18: Staffing Records Section

Department Head dashboard:

The Department Head, focuses on approval and acts as a decision authority after Resource Planners propose employees for projects or when employee assignments require departmental validation. From this dashboard, the Department Head reviews employee assignment requests along with contextual information such as project details, required capacity, timelines, and the employee's current workload. Based on this evaluation, the Department Head can approve or reject assignments. Once approved, the assignment proceeds to become active and visible in the Employee's "Assigned Projects" section and in Resource Planner staffing records; if rejected, the request is sent back to the Resource Planner or Project Manager for adjustment across the organization.

Employee Profiles Section:

The Employee Profiles section in the Department Head Dashboard enables the Department Head to monitor and manage all employees skills, availability, and current project assignments. The search functionality used to search employees by name, role, or skills and the availability status (All, Available, Not Available), which supports quick identification of suitable resources for upcoming or ongoing projects under the search section. Each employee profile shows information such as employee ID, contact details, role, base location, weekly capacity, skill set, and current project assignment status, clearly indicating whether the employee is available or already engaged.



Department Head Dashboard

Welcome back, Anita

Employee Profiles

View skills and availability of your team members

Tom Anderson (EmpId: 9)
tom.anderson@workforce.com
Backend Developer
Berlin
0 hours/week
SKILLS
Python Django REST API PostgreSQL
CURRENT PROJECT
No active project

Arun Malhotra (EmpId: 36)
arun.malhotra@workforce.com
Backend Developer
Frankfurt
40 hours/week
SKILLS
Java Spring Boot AWS REST API +1
CURRENT PROJECT
PRJ-IUV737

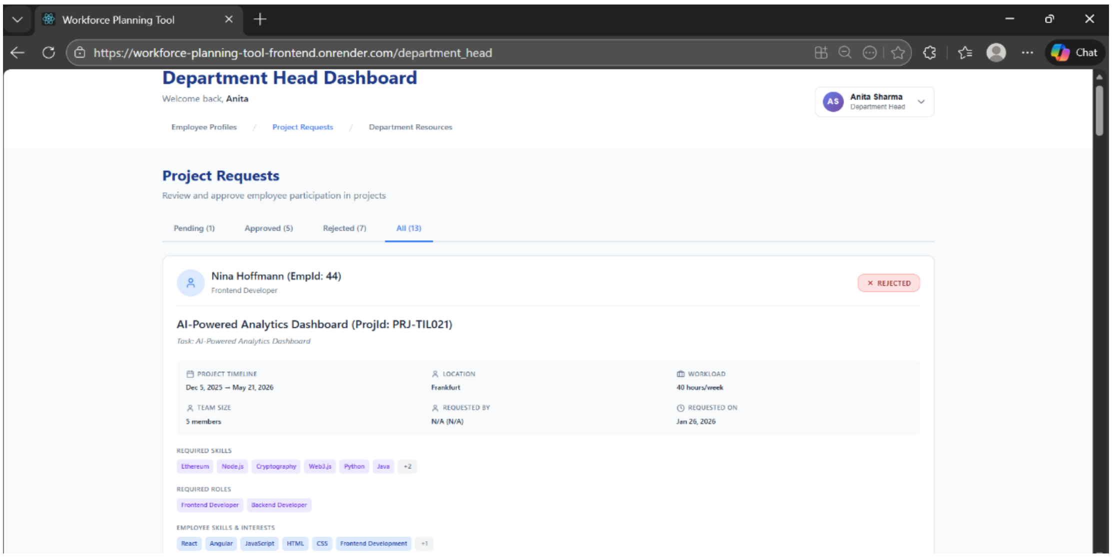
Noah Schneider (EmpId: 38)
noah.schneider@workforce.com
Cloud Operations Engineer
Munich
40 hours/week
SKILLS
Azure DevOps CI/CD Terraform Python +1
CURRENT PROJECT
PRJ-CJ5023

Figure 19: Employee Profiles Section

Project Requests Section:

The Project Requests section from Figure 20, allows the Department Head to review, approve, or reject employee requests for projects raised by Project Managers or generated by the system. This section serves as a key approval layer ensuring that employees are assigned to projects only after verifying capacity, skill alignment, and feasibility. It helps the Department Head maintain how team members are distributed across projects while aligning assignments with organizational priorities.

Requests are organized using tabs such as Pending, Approved, Rejected, and All, enabling quick navigation into the status of each request. Pending requests require immediate attention, while approved and rejected tabs provides a record of decisions made. This section provides detailed information, including employee details (name, employee ID, and role), project information (project name, project ID, and task description), project timeline, location, team size, and weekly workload. Additional context such as who requested the assignment, the request date, required skills and roles, and the employee's existing skills and interests allows the Department Head to check whether the employee is suitable and available for the requested project allocation.The Department Head can either approve or reject a request directly for the pending projects. Approving a request confirms the employee's participation and Rejecting a request prevents inappropriate assignments due to skill mismatch, capacity constraints, or departmental priorities.


Workforce Planning Tool
https://workforce-planning-tool-frontend.onrender.com/department_head
Chat
AS

Department Head Dashboard

Welcome back, Anita

Anita Sharma
Department Head

Employee Profiles / Project Requests / Department Resources

Project Requests

Review and approve employee participation in projects

Pending (1) Approved (5) Rejected (7) All (13)

Nina Hoffmann (EmpId: 44)
Frontend Developer

× REJECTED

AI-Powered Analytics Dashboard (ProjId: PRJ-TIL021)

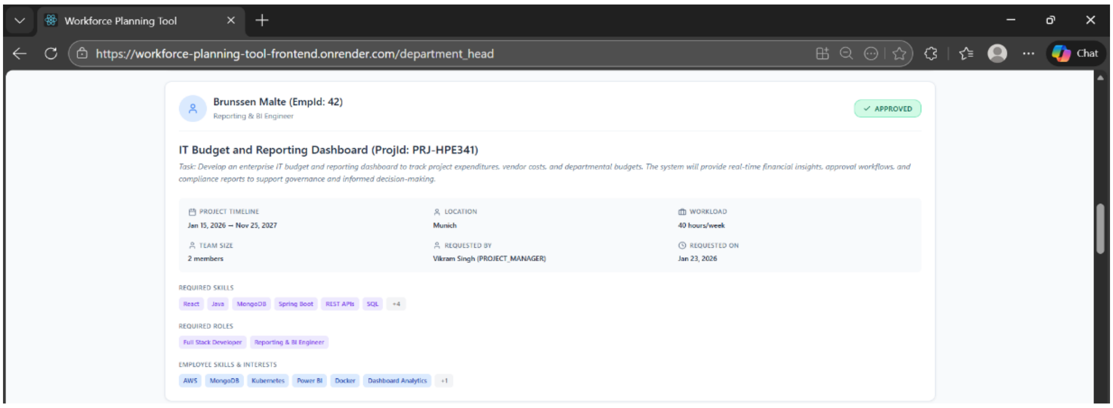
Task: AI-Powered Analytics Dashboard

PROJECT TIMELINE: Dec 5, 2025 → May 21, 2026
LOCATION: Frankfurt
WORKLOAD: 40 hours/week
TEAM SIZE: 5 members
REQUESTED BY: N/A (N/A)
REQUESTED ON: Jan 26, 2026

REQUIRED SKILLS
Ethereum, Node.js, Cryptography, Web3.js, Python, Java, +2

REQUIRED ROLES
Frontend Developer, Backend Developer

EMPLOYEE SKILLS & INTERESTS
React, Angular, JavaScript, HTML, CSS, Frontend Development, +1


Workforce Planning Tool
https://workforce-planning-tool-frontend.onrender.com/department_head
Chat

Brunssen Malte (EmpId: 42)
Reporting & BI Engineer

✓ APPROVED

IT Budget and Reporting Dashboard (ProjId: PRJ-HPE341)

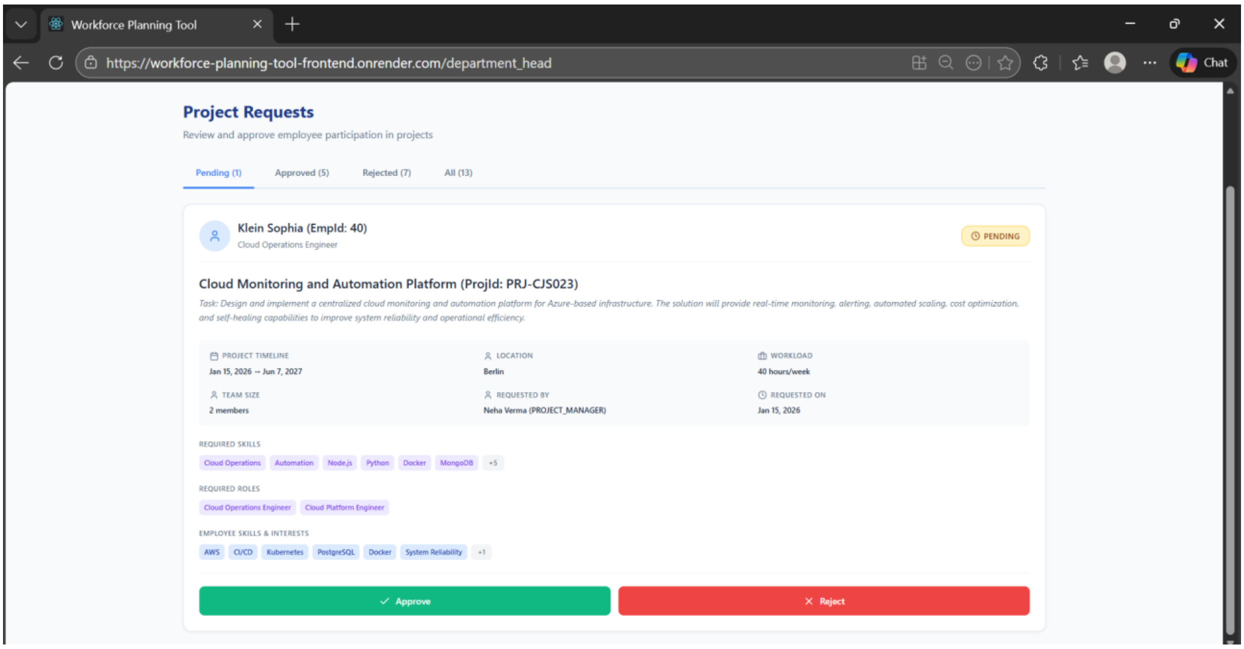
Task: Develop an enterprise IT budget and reporting dashboard to track project expenditures, vendor costs, and departmental budgets. The system will provide real-time financial insights, approval workflows, and compliance reports to support governance and informed decision-making.

PROJECT TIMELINE: Jan 15, 2026 → Nov 25, 2027
LOCATION: Munich
WORKLOAD: 40 hours/week
TEAM SIZE: 2 members
REQUESTED BY: Vikram Singh (PROJECT_MANAGER)
REQUESTED ON: Jan 23, 2026

REQUIRED SKILLS
React, Java, MongoDB, Spring Boot, REST APIs, SQL, +4

REQUIRED ROLES
Full Stack Developer, Reporting & BI Engineer

EMPLOYEE SKILLS & INTERESTS
AWS, MongoDB, Kubernetes, Power BI, Docker, Dashboard Analytics, +1


Workforce Planning Tool
https://workforce-planning-tool-frontend.onrender.com/department_head
Chat

Project Requests

Review and approve employee participation in projects

Pending (1) | Approved (5) | Rejected (7) | All (13)

Klein Sophia (EmpId: 40)
Cloud Operations Engineer

PENDING

Cloud Monitoring and Automation Platform (ProjId: PRJ-CJS023)

Task: Design and implement a centralized cloud monitoring and automation platform for Azure-based infrastructure. The solution will provide real-time monitoring, alerting, automated scaling, cost optimization, and self-healing capabilities to improve system reliability and operational efficiency.

PROJECT TIMELINE: Jan 15, 2026 → Jun 7, 2027
LOCATION: Berlin
WORKLOAD: 40 hours/week
TEAM SIZE: 2 members
REQUESTED BY: Neha Verma (PROJECT_MANAGER)
REQUESTED ON: Jan 15, 2026

REQUIRED SKILLS: Cloud Operations, Automation, Node.js, Python, Docker, MongoDB, +5

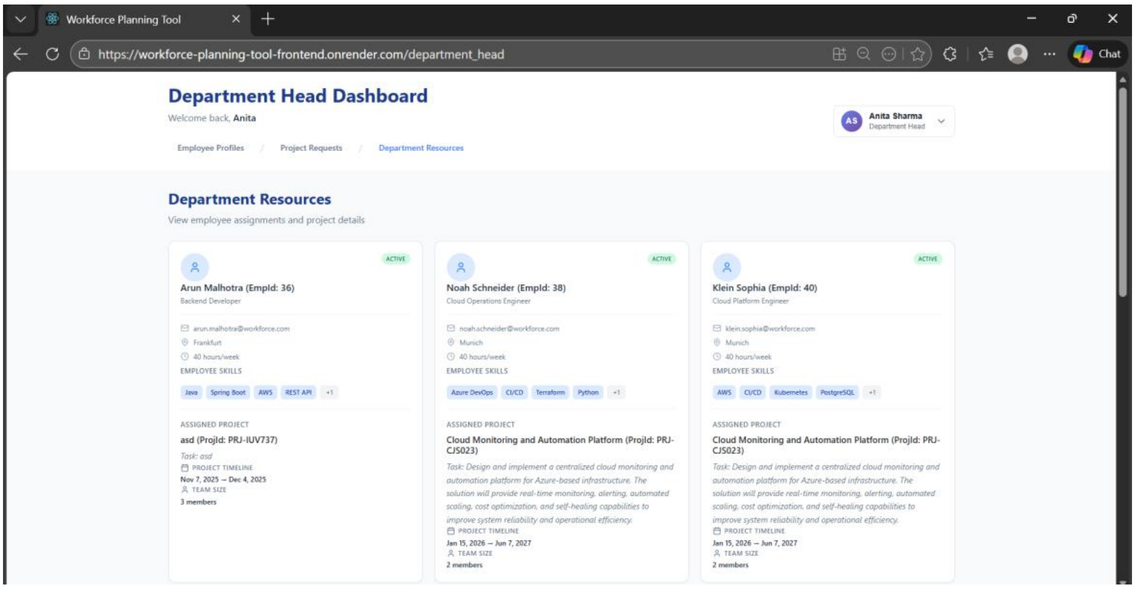
REQUIRED ROLES: Cloud Operations Engineer, Cloud Platform Engineer

EMPLOYEE SKILLS & INTERESTS: AWS, CI/CD, Kubernetes, PostgreSQL, Docker, System Reliability, +1

Figure 20: Project Requests Section

Department Resources Section:

The Department Resources section provides a view of employees are currently allocated across active projects. It enables the Department Head to monitor real time resource utilization by displaying each employee's assignment status, role, location, weekly capacity, and skill set and detailed assignment information, including the assigned project name and ID, task description, project timeline, and team size. By combining employee skills with their active project details, the Department Head can quickly assess whether current assignments align with employees' expertise and departmental objectives.


Workforce Planning Tool
https://workforce-planning-tool-frontend.onrender.com/department_head
Chat

Department Head Dashboard

Welcome back, Anita

AS Anita Sharma, Department Head

Employee Profiles / Project Requests / Department Resources

Department Resources

View employee assignments and project details

ACTIVE
Arun Malhotra (EmpId: 36)
Backend Developer
arun.malhotra@workforce.com
Frankfurt
40 hours/week
EMPLOYEE SKILLS: Java, Spring Boot, AWS, REST API, +1
ASSIGNED PROJECT
asd (ProjId: PRJ-IUV737)
Task: asd
PROJECT TIMELINE: Nov 7, 2025 – Dec 4, 2025
TEAM SIZE: 3 members

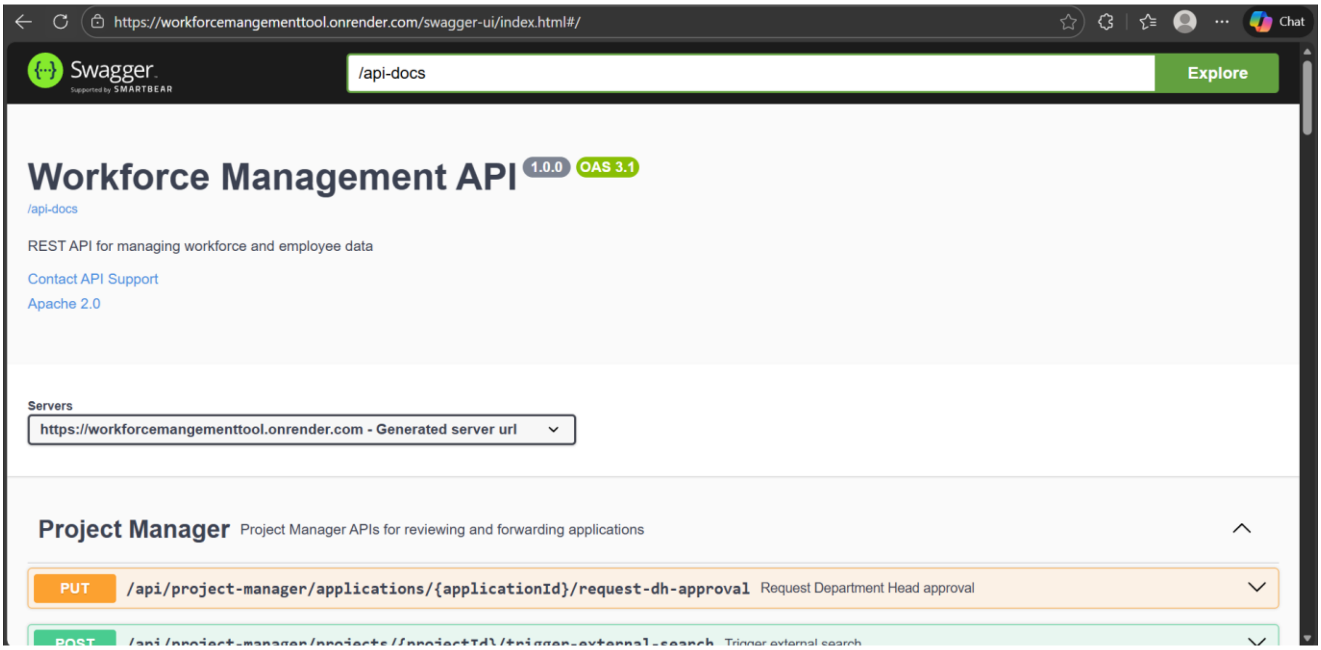
ACTIVE
Noah Schneider (EmpId: 38)
Cloud Operations Engineer
noah.schneider@workforce.com
Munich
40 hours/week
EMPLOYEE SKILLS: Azure DevOps, CI/CD, Terraform, Python, +1
ASSIGNED PROJECT
Cloud Monitoring and Automation Platform (ProjId: PRJ-CJS023)
Task: Design and implement a centralized cloud monitoring and automation platform for Azure-based infrastructure. The solution will provide real-time monitoring, alerting, automated scaling, cost optimization, and self-healing capabilities to improve system reliability and operational efficiency.
PROJECT TIMELINE: Jan 15, 2026 – Jun 7, 2027
TEAM SIZE: 2 members

ACTIVE
Klein Sophia (EmpId: 40)
Cloud Platform Engineer
klein.sophia@workforce.com
Munich
40 hours/week
EMPLOYEE SKILLS: AWS, CI/CD, Kubernetes, PostgreSQL, +1
ASSIGNED PROJECT
Cloud Monitoring and Automation Platform (ProjId: PRJ-CJS023)
Task: Design and implement a centralized cloud monitoring and automation platform for Azure-based infrastructure. The solution will provide real-time monitoring, alerting, automated scaling, cost optimization, and self-healing capabilities to improve system reliability and operational efficiency.
PROJECT TIMELINE: Jan 15, 2026 – Jun 7, 2027
TEAM SIZE: 2 members

Figure 21: Department Resources Section

API Documentation and Testing using Swagger:

Swagger (OpenAPI 3.1) is used to document and expose all REST APIs of the Workforce Management Tool in a structured and interactive manner. It provides a centralized interface where backend endpoints related to Project Manager, Department Head, Resource Planner, and Employee functionalities are clearly listed with HTTP methods, request parameters, and response formats. Developers and reviewers can directly explore and test APIs (such as approval requests and workflow actions) against the deployed Render server without additional tools. This improves API transparency, simplifies frontend–backend integration, and ensures consistent understanding of available services across the team.



https://workforcemangementtool.onrender.com/swagger-ui/index.html#/

/api-docs

Workforce Management API 1.0.0 OAS 3.1

/api-docs

REST API for managing workforce and employee data

Contact API Support

Apache 2.0

Servers

https://workforcemangementtool.onrender.com - Generated server url

Project Manager Project Manager APIs for reviewing and forwarding applications

PUT /api/project-manager/applications/{applicationId}/request-dh-approval Request Department Head approval

POST /api/project-manager/projects/{projectId}/trigger-external-search Trigger external search

Figure 22: Swagger API

Deployment:

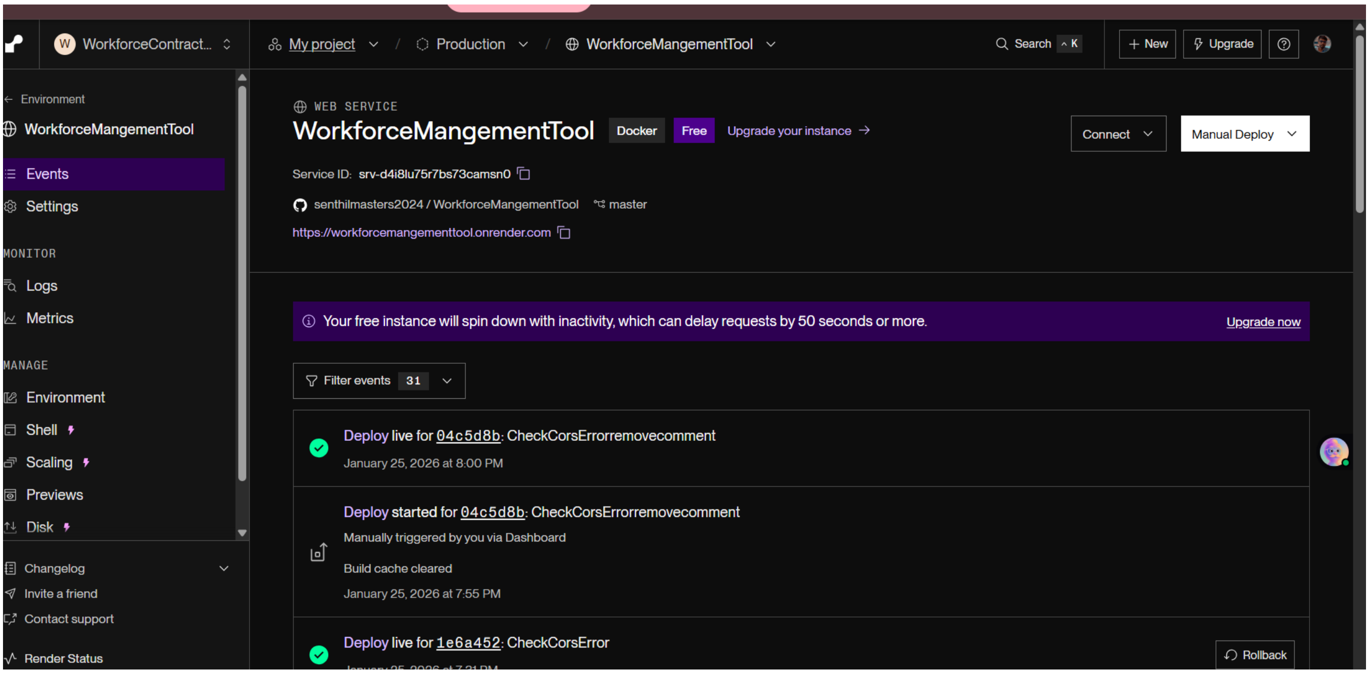
GitHub Link to Frontend: https://github.com/Babz-30/workforce-planning-tool-frontend
GitHub Link to Backend: https://github.com/senthilmasters2024/WorkforceMangementTool

The React application is deployed on Render using its static site hosting service. The production build is generated with npm run build, and the compiled assets from the build/ directory are served automatically by Render. Continuous deployment is enabled via a Git repository, allowing Render to rebuild and redeploy the application on every push to the main branch.

The backend services of the Workforce Management Tool are containerized using Docker with a multi-stage build approach to ensure efficiency and security. The build stage uses the gradle:8.5-jdk21 image to compile the application and generate an executable JAR using the bootJar task, while the runtime stage uses the lightweight eclipse-temurin:21-jre image. This approach reduces final image size and removes build tools from production. The Dockerfile is optimized for layer caching, skips test execution during image build (-x test), and exposes port 8080 for the Spring Boot application.

The application is deployed on Render as a managed Web Service, directly connected to the project's Git repository. Render automatically detects the Dockerfile and handles the build process. Runtime configuration is managed through environment variables, including the auto provided PORT, a secure MongoDB connection string, and SPRING_PROFILES_ACTIVE=prod. Application health is monitored using the /actuator/health endpoint, and deployment regions are selected to minimize latency.

For data persistence, the system uses MongoDB Atlas, a cloud-hosted NoSQL database, with the connection securely configured via environment variables such as SPRING_DATA_MONGODB_URI. The project follows a CI/CD workflow, where every push to the main branch triggers an automatic build and deployment on Render using rolling updates, ensuring zero downtime and reliable continuous delivery.



Service ID: srv-d4i8lu75r7bs73camsn0

senthilmasters2024 / WorkforceMangementTool master

https://workforcemangementtool.onrender.com

Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more.

Deploy live for 04c5d8b: CheckCorsErrorremovecomment
January 25, 2026 at 8:00 PM

Deploy started for 04c5d8b: CheckCorsErrorremovecomment
Manually triggered by you via Dashboard
Build cache cleared
January 25, 2026 at 7:55 PM

Deploy live for 1e6a452: CheckCorsError

Figure 23: Deployment

Comparison of technologies (Individual versus Camunda versus Appian versus Flowable - high level):

1. Individual Implementation (React + Spring Boot + Render):

The individual implementation using React, Spring Boot, and Render offers maximum flexibility and architectural control. React enables highly customizable, responsive dashboards tailored for different roles (Employee, Project Manager, Resource Planner, Department Head), while Spring Boot provides a robust backend for RESTful APIs, security, and business logic. This stack closely mirrors real-world enterprise development practices and supports scalable, cloud native deployment via Render with CI/CD integration. However, the approach requires significant development effort, manual workflow orchestration, explicit version control, and infrastructure monitoring. Unlike BPM platforms, business processes are embedded in code rather than visually modeled, making it harder for non-technical stakeholders to understand or modify workflows.

2. Camunda (BPMN-based Workflow Orchestration):

Camunda was used to decouple business processes from application code by modeling workflows using BPMN as a single source of truth. This approach improves process transparency, governance, and auditability, which is especially valuable in approval-heavy workflows such as project assignments and department head approvals. Camunda integrates well with Spring Boot and supports REST-based orchestration, but it introduces operational complexity, particularly around process correlation and monitoring. If correlation IDs are mismanaged, process instances can become unreachable. While Camunda excels in process integrity and standards compliance, it requires strong backend discipline and is less suitable for rapid UI-centric prototyping compared to low-code platforms.

3. Appian (Low-Code BPM Platform):

Appian is a full-stack low-code BPM platform designed for rapid workflow development and stakeholder collaboration. It provides built-in process modeling, UI forms, data handling, role management, and automatic versioning, allowing teams to prototype and deploy workflows quickly with minimal coding. This makes it highly suitable for business-driven applications where speed and governance are priorities. However, Appian's Community Edition restricts advanced features such as complex data modeling (CDTs), and its UI customization and REST extensibility are limited compared to developer-centric stacks. This reduces flexibility for highly customized, modern dashboards like those in your project.

4. Flowable (Lightweight BPM Engine):

Flowable is a developer-friendly BPM engine with strong support for BPMN and CMMN standards and seamless embedding into Spring Boot applications. It provides flexible task management, configurable user/service tasks, and a comprehensive REST API that works well with custom frontends like React. Compared to Appian, Flowable offers greater control and extensibility, but less out-of-the-box tooling. Debugging, monitoring, and operational visibility require more manual effort, as Flowable lacks the polished operational dashboards found in commercial BPM suites.

Aspect	**Individual Stack (React + Spring Boot + Render)**	**Camunda**	**Appian**	**Flowable**
Type	Full-stack custom development	BPMN-based workflow engine	Low-code BPM platform	Lightweight BPM engine
Primary Focus	UI flexibility, custom business logic, full system control	Process orchestration and workflow governance	Rapid workflow and app development	Embedded BPM with developer control
UI Development	Fully custom UI using React; modern and highly flexible	UI handled externally (custom frontend required)	Built-in UI components; limited customization	External UI (e.g., React) required
Backend Logic	Custom REST APIs and business logic in Spring Boot	Process logic modeled in BPMN, executed by engine	Logic defined within platform	Process logic modeled in BPMN/CMMN

Workflow Modeling	Hard-coded in backend logic	Visual BPMN models	Visual process models	Visual BPMN/CMMN models
Process Transparency	Low (code-centric)	Very high (BPMN diagrams)	High	High
Versioning & Rollback	Manual (Git, CI/CD pipelines)	Built-in process versioning	Built-in process versioning	Limited; mostly manual
Integration Capabilities	Strong REST, database, and third-party integration	Strong REST integration with backend services	Limited REST exposure, platform-dependent	Strong REST APIs and Java delegates
Learning Curve	High (frontend + backend + deployment)	Medium–high (BPMN + orchestration concepts)	Low (business-friendly)	Medium
Scalability	High (cloud-native, microservice-ready)	High	High (platform-managed)	High
Deployment	Cloud-native via Render	Requires engine deployment & management	Platform-managed	Embedded or standalone deployment
Monitoring & Debugging	Manual setup (logs, metrics, alerts)	Good but requires careful correlation handling	Built-in monitoring	Limited operational tooling
Cost & Licensing	Open-source stack, low cost	Open-source (enterprise features paid)	Proprietary, license-based	Open-source
Best Use Case	Custom enterprise applications with rich UI and logic	Complex approval workflows & governance	Rapid business process apps	BPM-driven apps with custom UI

Table 1: Comparison of Technologies

Conclusion

The Workforce Planning Tool not only fulfills academic expectations but serves as a production ready foundation for enterprise workforce management, demonstrating the practical application of Agile methodologies and modern system architecture in solving complex organizational challenges successfully and also meets the objectives defined in the project requirements by providing a role based, end to end solution for managing projects, resources, and workforce allocation within an organization. By clearly separating responsibilities across Project Managers, Resource Planners, Department Heads, and Employees, the system ensures transparency, accountability, and structured decision making throughout the project lifecycle. The implemented dashboards and approval flows reflect real world enterprise processes, enabling efficient collaboration and reducing manual coordination. From a technical perspective, the use of a custom built stack with React, Spring Boot, and cloud deployment allowed the team to design highly flexible user interfaces, implement tailored business logic, and integrate approval workflows that align closely with the use case specifications. Overall, the project demonstrates a strong understanding of Agile principles, system design, and application development. Despite challenges related to manual workflow implementation and coordination across multiple roles, the final solution delivers a robust, well-structured workforce planning platform. The project not only fulfills academic expectations but also closely resembles a real world enterprise application, making it a solid foundation for further enhancement and practical deployment.