AGILE Development In Cloud Computing Environments

Service Management Component

Masters in Engineering
Information Technology

Project Members

1.	Farima Javadi	1428759
2.	Kavya Hirebelaguli Chandrashekar	1429305
3.	Pranil Prakash Ghadi	1441215
4.	Richa Padhi	1438260
5.	Soheil Nasiri	1324594
6.	Theertha Bharathan	1445457

Introduction

- **Overview of the Provider Management Platform**

This project aims to develop a Provider Management Platform (PMP) utilizing Agile principles. The platform serves as a space where job availabilities, along with detailed project descriptions, are shared. Providers, representing external companies with suitable employees, register and log in to view job availabilities. Providers enter details for selected profiles, awaiting evaluation by the portal's staff. Based on this evaluation, positive results lead to the issuance of offers, while negative outcomes pass provider data to new projects.

- **Service Management Component**
  The Service Management Component is a critical module within the Provider Management Platform, focusing on the efficient management of IT service requests. This component enables users within the organization to initiate, track, and manage service requests for various IT services. It incorporates features such as creating service requests, evaluating offers from providers, and assessing the quality of services delivered. Additionally, the Service Management Component facilitates collaboration with providers and allows users to provide feedback on service delivery.

  Key features of the Service Management Component include:

  - **Service Request Creation:** Users can initiate service requests, including multi-requests and team requests, specifying details such as expertise level and required roles.

  - **Request Status Tracking:** Users can monitor the status of the service requests they initiated, providing transparency and accountability in the workflow.

  - **Offer Evaluation:** The component allows users to evaluate offers received from providers in response to service requests, enabling informed decision-making.

  - **Feedback:** The component allows users to provide the reason behind the rejection of an offer

Agile Methodology & its Principles

Agile is a flexible and efficient project management approach designed to bring products to market quickly. Agile projects are characterized by short intervals or sprints, allowing frequent adjustments based on requirements. The Agile process emphasizes pulling tasks rather than pushing, promoting faster adaptation and turnaround for workable products.

Kanban Framework

Kanban, a Lean workflow management method, is employed in developing the Provider Management Platform. It visualizes work, maximizes efficiency, and improves continuously. User stories, representing project requirements, are assigned to developers and managed on a Kanban Board. This allows optimization of work delivery and handling complex projects in a unified environment.

SCRUM as Agile

Scrum, another agile framework, structures and manages work through principles and practices. It breaks projects into manageable "Sprints." The Scrum master plays a pivotal role in understanding engineers' availability, clarifying project requirements, and organizing user stories into sprints. Frequent meetings with the customer and integrations with different teams ensure continuous progress.

Database

Data is stored using PostgreSQL, a robust relational database management system known for its reliability and scalability in building highly available internet applications. PostgreSQL's structured and relational data model supports agile development by providing a solid foundation for immediate application building without the need for extensive database configuration. Its adherence to a predefined schema ensures data integrity and facilitates efficient querying, contributing to a seamless development process.

Sprint Planning

Sprint Planning is a crucial event in Scrum, defining deliverables for the upcoming sprint and outlining the work plan. The steps involved in Sprint Planning include:

1. Reminding the team of the big picture or goal.
2. Discussing any new information impacting the plan.
3. Confirming team capacity and addressing known issues.
4. The Product Owner answering questions and elaborating acceptance criteria.
5. This process ensures a clear understanding of project goals and efficient planning for each sprint.

Sprint 1 (Length- 2 week)

During this sprint, the team received information about the technology chosen by the client, Professor Wacht. The project requirements were outlined and distributed. Following the Agile principle, we acknowledged the likelihood of changes, and occasionally, a given requirement might be shifted to the backlog for the next sprint. With the requirements established, our focus shifted to project planning. To facilitate effective management, the requirements were divided into user stories, each assigned to developers with an estimated time frame for completion.

Sprint 2 (Length- 2 week)

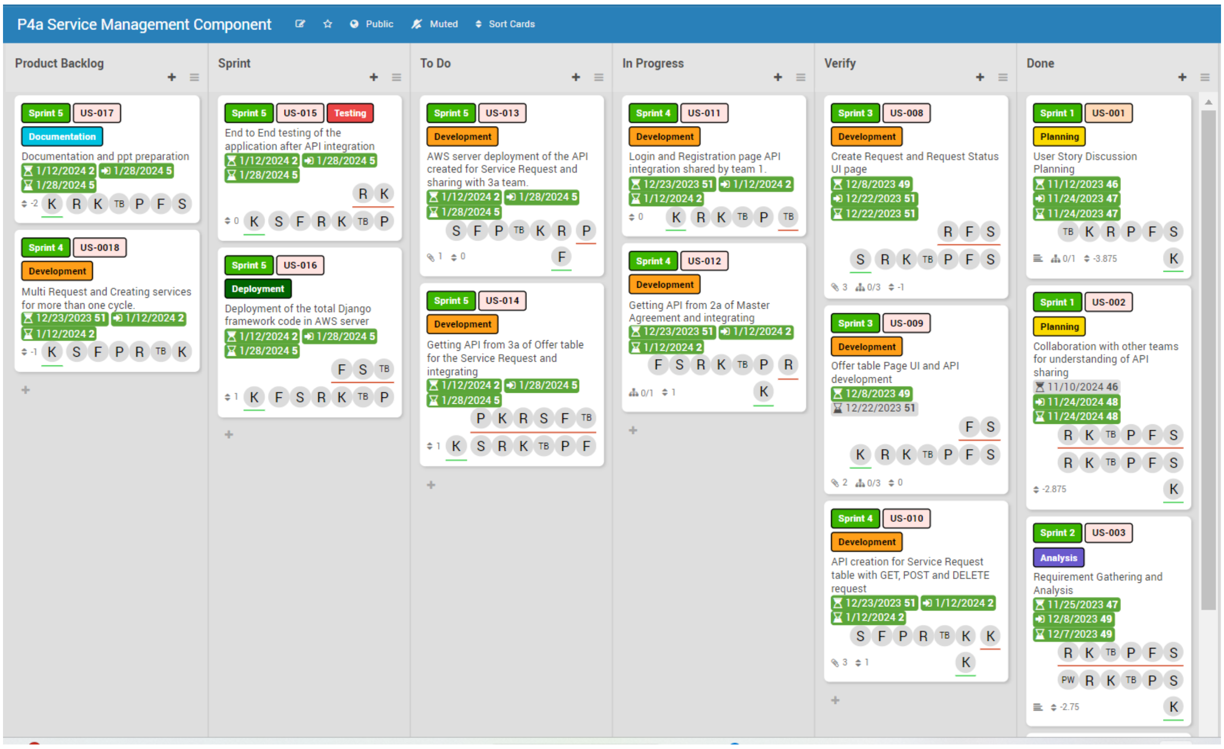
Sprint 2 mainly focused on designing how the software will be built. Deciding the structure of the system, designing the database, API, user interface based on user requirement and figuring out how different parts of the software will work together. Building upon this foundational work, the team has started focused on the implementation of the service request functionality. The team concentrated on developing an intuitive user interface to allow users to create service requests for IT services.

Sprint 3 (Length- 2 week)

Sprint 3 was focused on enhancing the capabilities of the Service Management Component. Users were empowered with the capability to create a dedicated page for showcasing all the service requests they have generated. Additionally, a delete button feature was implemented, allowing users to remove requests that have not received any offers. The status of each request became visible to users as part of these advancements. Also developing the offers page UI.

Sprint 4 (Length- 2 weeks)

In Sprint 4, our team achieved significant milestones in API development and integration. We successfully created APIs for the Service Request table, enabling seamless interaction with the system through GET, POST, and DELETE requests. Additionally, we collaborated with Team 1 to integrate their Login and Registration page APIs, enhancing our system's user authentication capabilities. Moreover, we expanded the scope of our application by incorporating APIs from the 2a section of the Master Agreement, facilitating the retrieval and integration of pertinent data.



P4a Service Management Component

Public · Muted · Sort Cards

Product Backlog

- Sprint 5 · US-017 · Documentation — Documentation and ppt preparation — 1/12/2024 2 · 1/28/2024 5 · 1/28/2024 5
- Sprint 4 · US-0018 · Development — Multi Request and Creating services for more than one cycle. — 12/23/2023 51 · 1/12/2024 2 · 1/12/2024 2

Sprint

- Sprint 5 · US-015 · Testing — End to End testing of the application after API integration — 1/12/2024 2 · 1/28/2024 5 · 1/28/2024 5
- Sprint 5 · US-016 · Deployment — Deployment of the total Django framework code in AWS server — 1/12/2024 2 · 1/28/2024 5 · 1/28/2024 5

To Do

- Sprint 5 · US-013 · Development — AWS server deployment of the API created for Service Request and sharing with 3a team. — 1/12/2024 2 · 1/28/2024 5 · 1/28/2024 5
- Sprint 5 · US-014 · Development — Getting API from 3a of Offer table for the Service Request and integrating — 1/12/2024 2 · 1/28/2024 5 · 1/28/2024 5

In Progress

- Sprint 4 · US-011 · Development — Login and Registration page API integration shared by team 1. — 12/23/2023 51 · 1/12/2024 2 · 1/12/2024 2
- Sprint 4 · US-012 · Development — Getting API from 2a of Master Agreement and integrating — 12/23/2023 51 · 1/12/2024 2 · 1/12/2024 2

Verify

- Sprint 3 · US-008 · Development — Create Request and Request Status UI page — 12/8/2023 49 · 12/22/2023 51 · 12/22/2023 51
- Sprint 3 · US-009 · Development — Offer table Page UI and API development — 12/8/2023 49 · 12/22/2023 51
- Sprint 4 · US-010 · Development — API creation for Service Request table with GET, POST and DELETE request — 12/23/2023 51 · 1/12/2024 2 · 1/12/2024 2

Done

- Sprint 1 · US-001 · Planning — User Story Discussion Planning — 11/12/2023 46 · 11/24/2023 47 · 11/24/2023 47
- Sprint 1 · US-002 · Planning — Collaboration with other teams for understanding of API sharing — 11/10/2024 46 · 11/24/2024 48 · 11/24/2024 48
- Sprint 2 · US-003 · Analysis — Requirement Gathering and Analysis — 11/25/2023 47 · 12/8/2023 49 · 12/7/2023 49

Fig.1 Kanban Board

Sprint 5 (Length- 2 weeks)

Sprint 5 marked a crucial phase where the team concentrated on optimizing user interactions and implementing the offer evaluation process. The team incorporated a pivotal task - integration with APIs from 3a. The implementation included the ability to display multiple offers for each request, accompanied by options to accept or decline. As part of the offer evaluation process, a feedback mechanism was introduced to provide insights in case an offer was declined.

Additionally, comprehensive End-to-End testing was conducted for the entire application after completing API integration. This thorough testing ensures that all components work together seamlessly, identifying and addressing any potential issues. At the end we successfully deployed the complete Django framework code on an AWS server. This deployment represents a crucial step in making the application accessible to users, demonstrating progress towards the project's final implementation.

Sprint Meetings & Review

Sprint Meetings and Reviews play a crucial role in the Agile methodology, providing opportunities for collaboration, assessment, and adaptation throughout the development process. In these iterative cycles, the project team comes together to discuss progress, and challenges, and plan for the next steps. Sprint Meetings typically commence each sprint and are characterized by the following components:

- Sprint Planning: At the beginning of the sprint, the team collaboratively plans and prioritizes tasks, determining what can be achieved during the upcoming iteration.
- Daily Stand-ups: Short, daily meetings where team members share updates on their work, discuss impediments, and coordinate efforts. These brief sessions foster communication and keep the team aligned.

- Sprint Review: At the end of each sprint, a review session is conducted to showcase the completed work to stakeholders. This session includes a demonstration of the deliverables, facilitating feedback and adjustments.
- Sprint Retrospective: Following the review, the team reflects on the sprint, discussing what went well, what could be improved, and identifying action items for continuous enhancement.

Sprint Meetings ensure transparency, effective communication, and alignment among team members. They enable the team to adapt to changing requirements, address challenges, and continuously deliver value. Regular reviews during these sprints ensure that the project aligns with stakeholder expectations and remains on track for success.

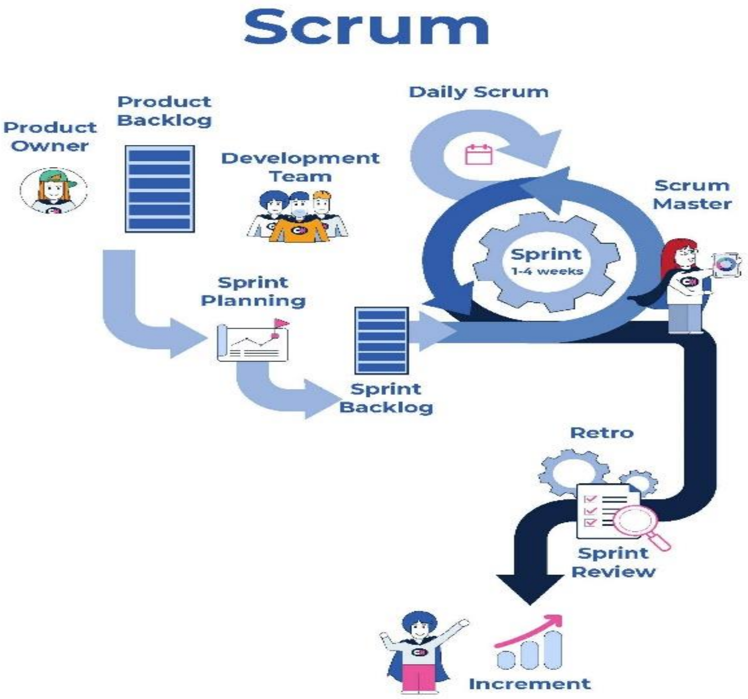


Fig 2. Sprint Meetings and Reviews

Push and Pull Principle

1. Code Collaboration and Version Control:

- Push Principle:

- Team members push their completed code changes to the shared GitHub repository after completing their assigned tasks.
- This follows the agile principle of delivering working software frequently, ensuring that increments of value are regularly integrated into the codebase.
- Pull Principle:
- Before starting new tasks or making further changes, team members pull the latest code from the shared repository.
- This aligns with agile principles by promoting continuous integration and collaboration, allowing team members to work on an up-to-date codebase.

2. Collaboration and Communication:

- Communication on Code Changes:

- Team members communicate their completed tasks and push code to the team, fostering transparency and ensuring everyone is aware of the progress.
- This practice aligns with the agile principle of individuals and interactions over processes and tools.
- Handling Concurrent Changes:
- In case two people are working on the same piece of code, proper communication takes place to avoid conflicts.
- This reflects the agile principle of responding to change over following a plan, as the team adapts to ensure collaboration and synchronization.

3. Flexibility and Adaptability:

- Adapting to Changes:

- The push and pull technique allow the team to adapt quickly to changes. If conflicts arise, the team can communicate and resolve them promptly.
- This embodies the agile principle of embracing change for the customer's competitive advantage.

4. Continuous Integration:

- Automated Builds and Tests:

- Continuous integration is supported by automated builds and tests triggered upon each push, ensuring the code remains in a deployable state.
- This aligns with the agile principle of delivering a potentially shippable product increment at the end of each iteration.

5. Empowering Individuals and Teams:

- Individual Contributions:

- Each team member is empowered to work independently on their tasks, making progress at their own pace.
- This reflects the agile principle of trusting motivated individuals and providing them with the environment and support they need.

6. Feedback and Inspection:

- Pull Requests/Code Reviews:

- Pull requests and code reviews are used to inspect and provide feedback on code changes before merging into the main branch.
- This aligns with the agile principle of reflecting regularly on effectiveness and adjusting behavior accordingly.

Opinion on Agile Principles

Positive Aspects:

1. Collaboration and Communication: Team members appreciate the emphasis on regular communication and collaboration. Agile ceremonies such as Daily Standups and Sprint Reviews create opportunities for open discussions and quick problem-solving.

2. Adaptability: The ability to adapt to changing requirements and priorities is seen as a significant advantage. Agile's iterative approach allows teams to respond quickly to feedback and evolving customer needs.

3. Customer Focus: Agile principles encourage a customer-centric approach. Regular interactions with stakeholders, frequent feedback loops, and the delivery of incremental value contribute to a strong customer focus.

4. Continuous Improvement: Teams value the iterative nature of agile methodologies, including regular retrospectives. This allows them to reflect on their processes, celebrate successes, and identify areas for improvement.

5. Empowerment: Agile principles emphasize trusting and empowering individuals. Team members appreciate the autonomy and responsibility they have in delivering high-quality work.

Challenges and Considerations:

1. Learning Curve: Transitioning to agile methodologies may pose a learning curve for some team members, especially those accustomed to traditional project management approaches.

2. Documentation: Some team members may express concerns about the minimal emphasis on extensive documentation in agile methodologies. Balancing documentation needs with agility is crucial.

3. Team Dynamics: Successful application of agile principles requires strong team collaboration. Teams may face challenges if there are communication gaps or if individuals resist a more collaborative approach.

4. Rigidity of Some Practices: Depending on the specific agile framework being used, teams may find certain practices too rigid. Flexibility in tailoring agile processes to suit the team's unique context is important.

Overall, the general opinion is likely to be positive, with teams recognizing the benefits of agile methodologies in fostering collaboration, adaptability, and a customer-oriented mindset. Continuous improvement, open communication, and a commitment to delivering value are key aspects contributing to a favorable view of agile principles. Teams need to embrace agility not as a set of rigid rules but as a set of guiding principles that can be adapted to suit their specific needs and challenges.

Importance of Product Owner

The necessity of having the roles of Product Owner and Scrum Master in a Scrum team depends on the specific context, project requirements, and the team's dynamics. However, these roles are integral to the Scrum framework and serve distinct purposes:

- The Product Owner is crucial for maintaining a clear product vision, making informed decisions, and ensuring that the team delivers value aligned with customer needs.
- Having a dedicated Product Owner helps in avoiding conflicting priorities and provides a single point of contact for clarifications on requirements.

Service Management Component

The Platform is available at: the URL address here
This platform serves as a central hub creating service requests based on required domain and roles.

Service Management Platform Flowchart

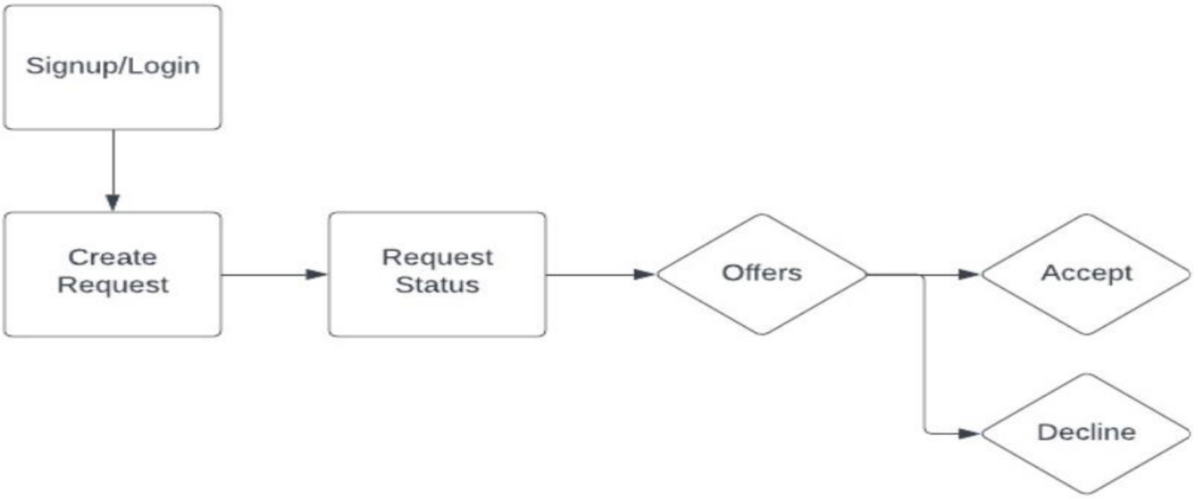


Fig 3: Flowchart

Home Page

Our home page serves as the gateway to Service management, providing a concise overview of the Create Request, Request Status, and Offers.

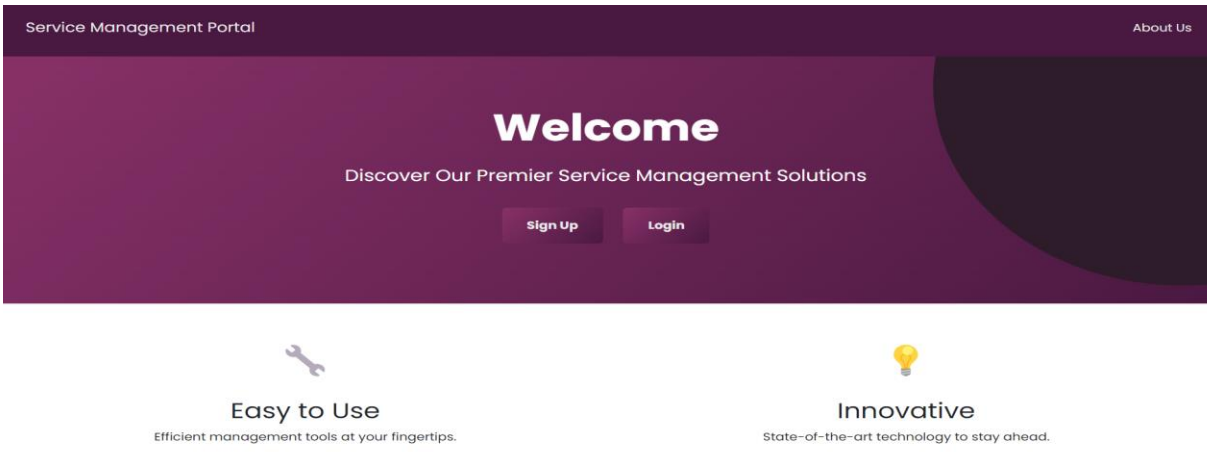

Service Management Portal
About Us
Welcome
Discover Our Premier Service Management Solutions
Sign Up
Login
Easy to Use
Efficient management tools at your fingertips.
Innovative
State-of-the-art technology to stay ahead.

Fig 4: Home Page

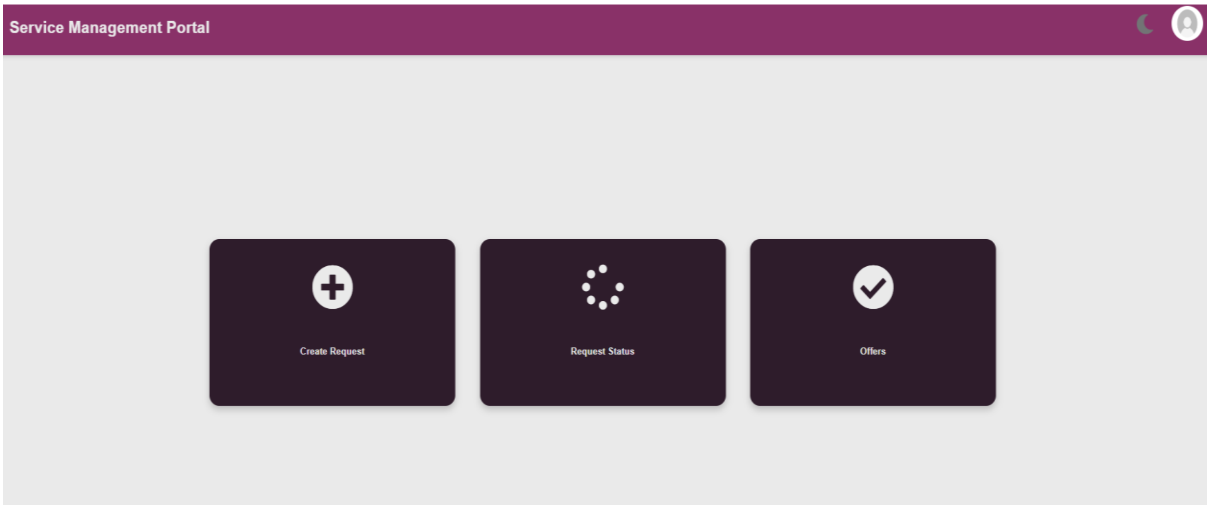
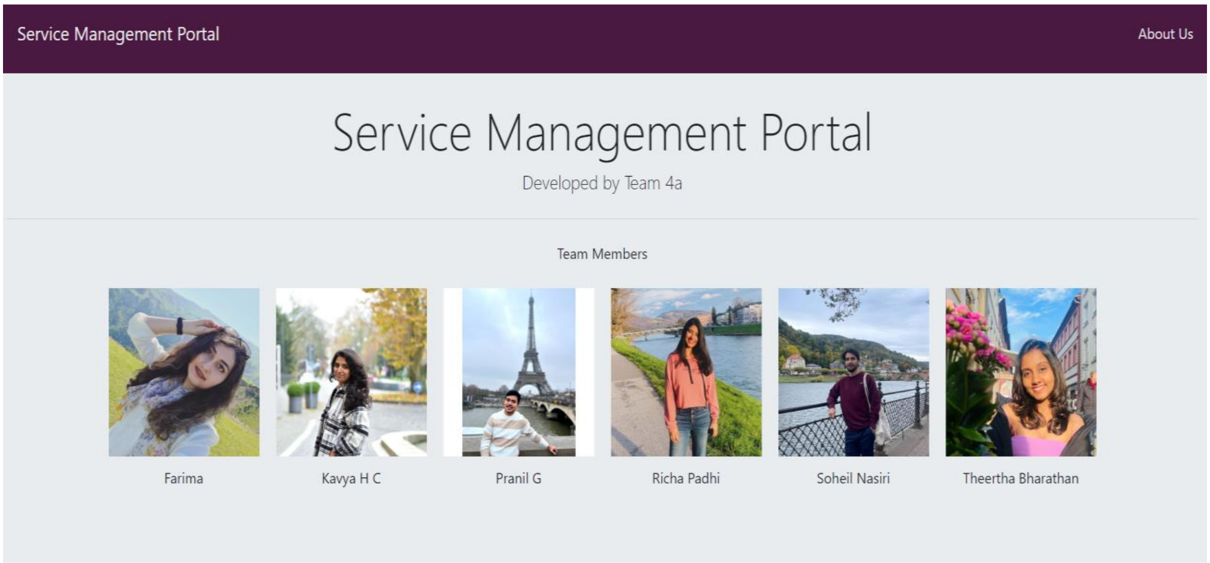


Fig 5: Dashboard

Curious to know who's driving the innovation behind the scenes? Our "About Us" section sheds light on the engineers and contributors responsible for shaping the Page.



Service Management Portal

Developed by Team 4a

Team Members

Farima

Kavya H C

Pranil G

Richa Padhi

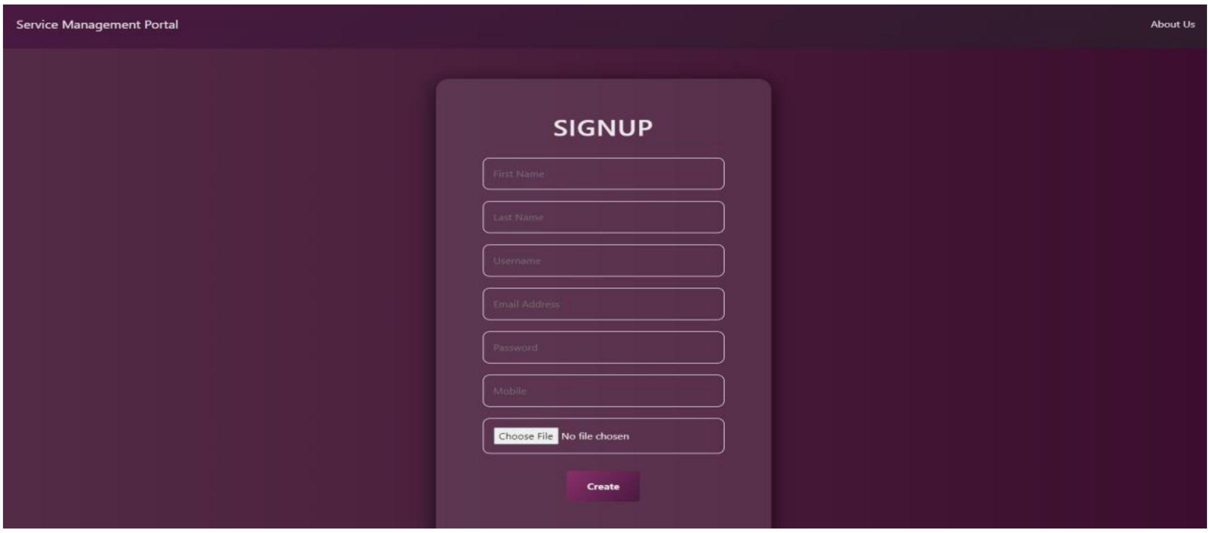
Soheil Nasiri

Theertha Bharathan

Fig 6: About Us

Register /Login Page

Accessing Service Management Page is a streamlined process through our Registration and Login Page. To participate in the platform, clients are required to register and log in. The registration page offers a straightforward sign-up experience, featuring fields for username, email, password, and a category field. A single click on the 'Sign Up' button initiates the registration process. This has been done by integrating with API we received from Group 1, where the credentials are stored in their database.



SIGNUP

First Name

Last Name

Username

Email Address

Password

Mobile

Choose File No file chosen

Fig 7: Signup page(integration with group 1)

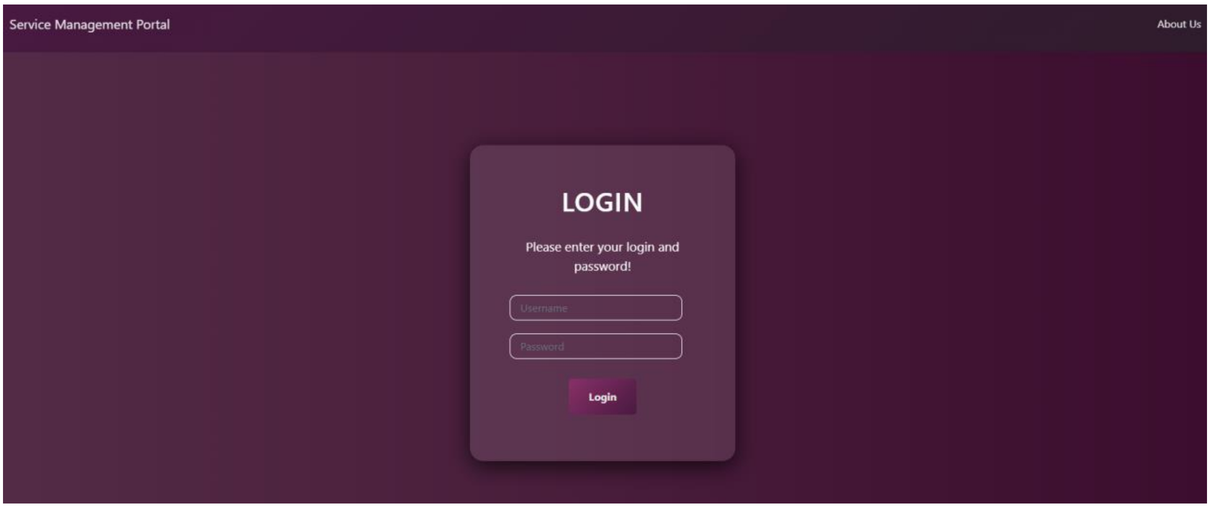

Service Management Portal
About Us
LOGIN
Please enter your login and password!
Username
Password
Login

Fig 8: Login page(integration with group 1)

Create Request

The "Create Request" page in this project's Service Management Component serves as a user-friendly interface enabling providers to initiate various types of service requests for IT services. Users can specify the nature of their requests. The page facilitates the inclusion of detailed requirements, such as desired expertise levels and specific roles needed for the requested services. We retrieve master agreements from Team 2a by utilizing their API, specifically using the masteragreementtype name and masteragreementid. Subsequently, based on the Master Agreement Type (MAT), we proceed to extract information such as Domain and Role associated with each MAT.

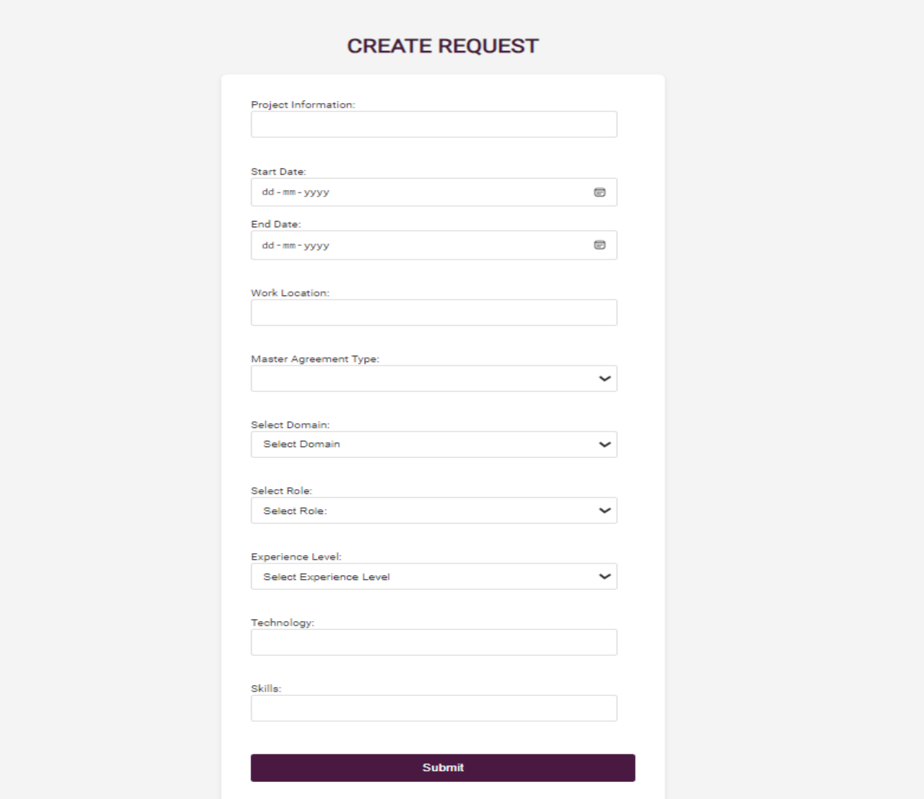
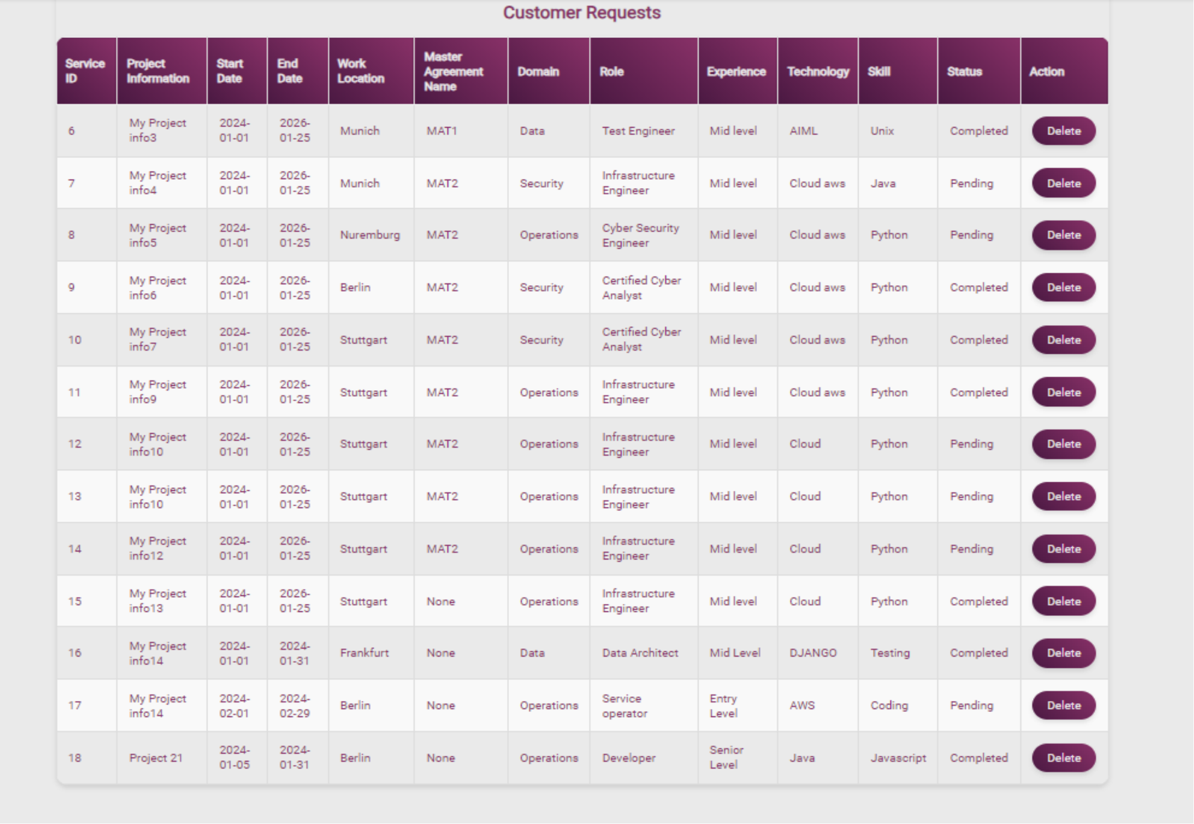


Fig 9: Create Request

Request Status

The Request Status Page functionality allows users to conveniently monitor the status of requests they initiated. Users can easily track the progress of their requests and, if needed, exercise control by deleting requests directly through the status page but only if an offer has not been provided for the respective request. This feature provides users with real-time visibility into the lifecycle of their requests and empowers them with the ability to manage and modify their submissions as necessary, enhancing overall user experience and engagement with the system.



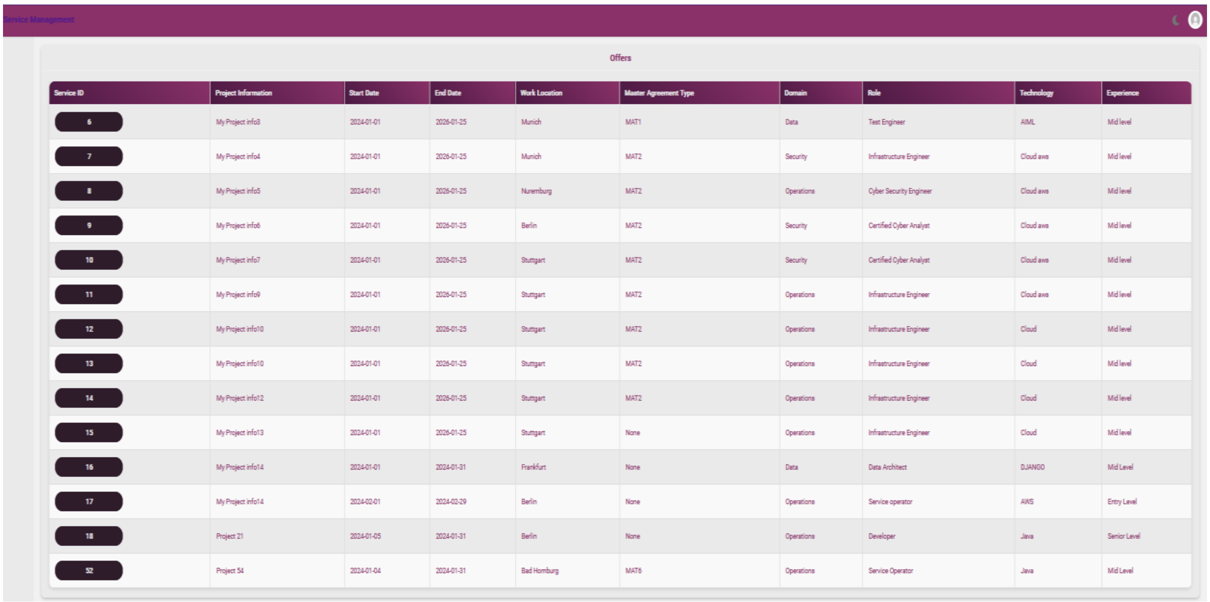
Customer Requests

Service ID	Project Information	Start Date	End Date	Work Location	Master Agreement Name	Domain	Role	Experience	Technology	Skill	Status	Action
6	My Project info3	2024-01-01	2026-01-25	Munich	MAT1	Data	Test Engineer	Mid level	AIML	Unix	Completed	Delete
7	My Project info4	2024-01-01	2026-01-25	Munich	MAT2	Security	Infrastructure Engineer	Mid level	Cloud aws	Java	Pending	Delete
8	My Project info5	2024-01-01	2026-01-25	Nuremburg	MAT2	Operations	Cyber Security Engineer	Mid level	Cloud aws	Python	Pending	Delete
9	My Project info6	2024-01-01	2026-01-25	Berlin	MAT2	Security	Certified Cyber Analyst	Mid level	Cloud aws	Python	Completed	Delete
10	My Project info7	2024-01-01	2026-01-25	Stuttgart	MAT2	Security	Certified Cyber Analyst	Mid level	Cloud aws	Python	Completed	Delete
11	My Project info9	2024-01-01	2026-01-25	Stuttgart	MAT2	Operations	Infrastructure Engineer	Mid level	Cloud aws	Python	Completed	Delete
12	My Project info10	2024-01-01	2026-01-25	Stuttgart	MAT2	Operations	Infrastructure Engineer	Mid level	Cloud	Python	Pending	Delete
13	My Project info10	2024-01-01	2026-01-25	Stuttgart	MAT2	Operations	Infrastructure Engineer	Mid level	Cloud	Python	Pending	Delete
14	My Project info12	2024-01-01	2026-01-25	Stuttgart	MAT2	Operations	Infrastructure Engineer	Mid level	Cloud	Python	Pending	Delete
15	My Project info13	2024-01-01	2026-01-25	Stuttgart	None	Operations	Infrastructure Engineer	Mid level	Cloud	Python	Completed	Delete
16	My Project info14	2024-01-01	2024-01-31	Frankfurt	None	Data	Data Architect	Mid Level	DJANGO	Testing	Completed	Delete
17	My Project info14	2024-02-01	2024-02-29	Berlin	None	Operations	Service operator	Entry Level	AWS	Coding	Pending	Delete
18	Project 21	2024-01-05	2024-01-31	Berlin	None	Operations	Developer	Senior Level	Java	Javascript	Completed	Delete

Fig 10: Request Status Page

Offers

The offer selection and evaluation feature enables users to choose profiles from offers that align with the requirements specified in their service requests. Users can then assess these offers by either accepting or declining them. In the event of declining an offer, users have the option to provide reasons for the rejection, offering valuable feedback to the system. This interactive process streamlines the user's decision-making process, fostering effective communication and enhancing the overall experience between service requestors and providers.



Offers

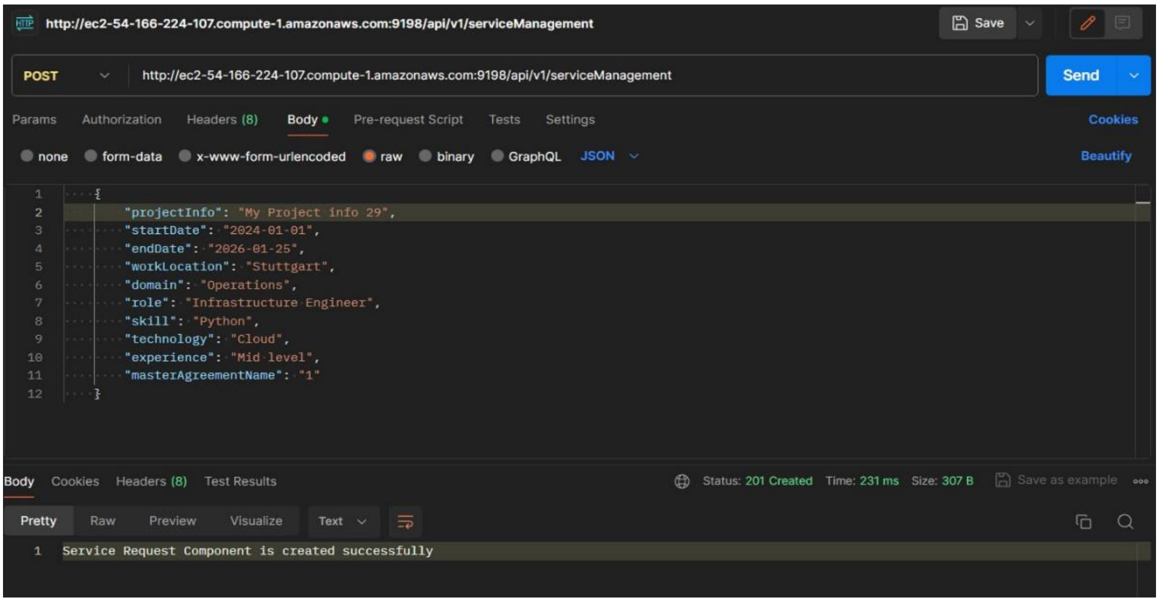
Service ID	Project Information	Start Date	End Date	Work Location	Master Agreement Type	Domain	Role	Technology	Experience
6	My Project info3	2024-01-01	2026-01-25	Munich	MAT1	Data	Test Engineer	AIML	Mid level
7	My Project info4	2024-01-01	2026-01-25	Munich	MAT2	Security	Infrastructure Engineer	Cloud aws	Mid level
8	My Project info5	2024-01-01	2026-01-25	Nuremburg	MAT2	Operations	Cyber Security Engineer	Cloud aws	Mid level
9	My Project info6	2024-01-01	2026-01-25	Berlin	MAT2	Security	Certified Cyber Analyst	Cloud aws	Mid level
10	My Project info7	2024-01-01	2026-01-25	Stuttgart	MAT2	Security	Certified Cyber Analyst	Cloud aws	Mid level
11	My Project info9	2024-01-01	2026-01-25	Stuttgart	MAT2	Operations	Infrastructure Engineer	Cloud aws	Mid level
12	My Project info10	2024-01-01	2026-01-25	Stuttgart	MAT2	Operations	Infrastructure Engineer	Cloud	Mid level
13	My Project info10	2024-01-01	2026-01-25	Stuttgart	MAT2	Operations	Infrastructure Engineer	Cloud	Mid level
14	My Project info12	2024-01-01	2026-01-25	Stuttgart	MAT2	Operations	Infrastructure Engineer	Cloud	Mid level
15	My Project info13	2024-01-01	2026-01-25	Stuttgart	None	Operations	Infrastructure Engineer	Cloud	Mid level
16	My Project info14	2024-01-01	2024-01-31	Frankfurt	None	Data	Data Architect	DJANGO	Mid Level
17	My Project info14	2024-02-01	2024-02-29	Berlin	None	Operations	Service operator	AWS	Entry Level
18	Project 21	2024-01-05	2024-01-31	Berlin	None	Operations	Developer	Java	Senior Level
52	Project 54	2024-01-04	2024-01-31	Bad Homburg	MAT6	Operations	Service Operator	Java	Mid Level

Fig 11: Offer Page

API of Service Management Component

POST

The API is available here. Using the POST method, when we complete the service request form, a new entry is generated in our database. The primary key for each record is the serviceID, which increments with each entry. All the information entered into the form is then sent and posted to the API of our database. This method facilitates the creation of new records and the seamless addition of information to our database.



POST http://ec2-54-166-224-107.compute-1.amazonaws.com:9198/api/v1/serviceManagement

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "projectInfo": "My Project info 29",
    "startDate": "2024-01-01",
    "endDate": "2026-01-25",
    "workLocation": "Stuttgart",
    "domain": "Operations",
    "role": "Infrastructure Engineer",
    "skill": "Python",
    "technology": "Cloud",
    "experience": "Mid level",
    "masterAgreementName": "1"
}
```

Body — Status: 201 Created Time: 231 ms Size: 307 B

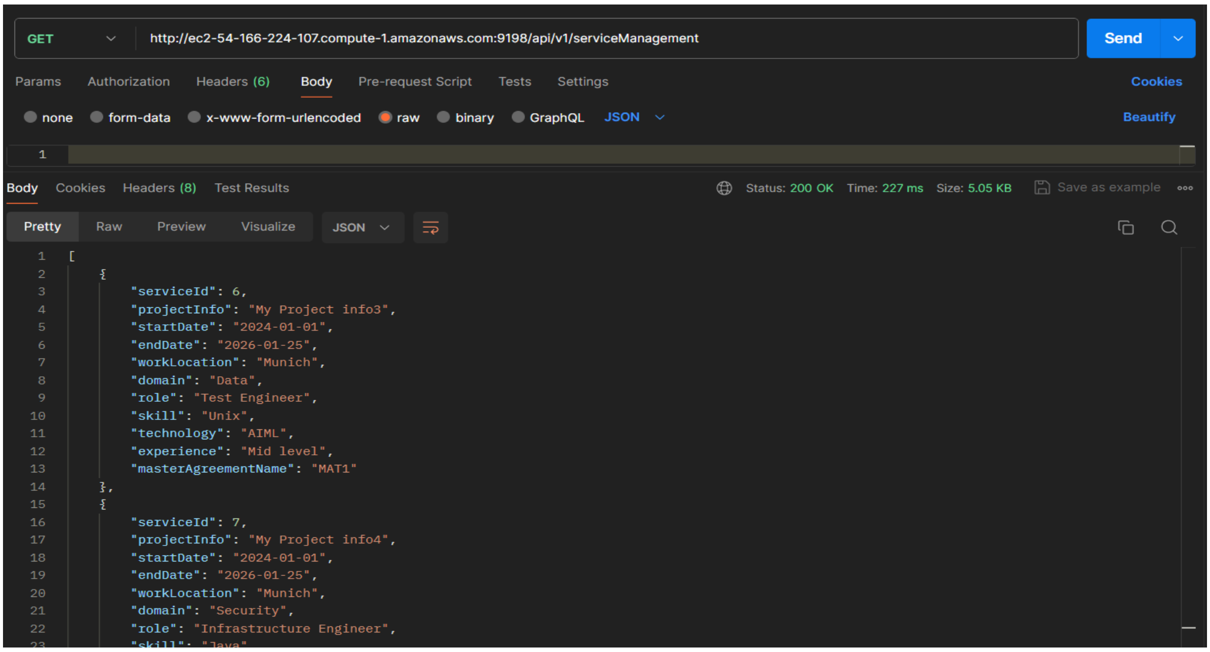
Text

```
Service Request Component is created successfully
```

Fig 12: API POST

GET

The API is available here. For the API creation for the Service Request table, the GET request allows users to retrieve information from the Service Request table based on specified parameters such as projectInfo, startDate, endDate, workLocation, domain, role, skill, technology, experience, and masterAgreementName. This retrieval operation is crucial for users seeking details about existing service requests, enabling them to access and review relevant information.



GET http://ec2-54-166-224-107.compute-1.amazonaws.com:9198/api/v1/serviceManagement

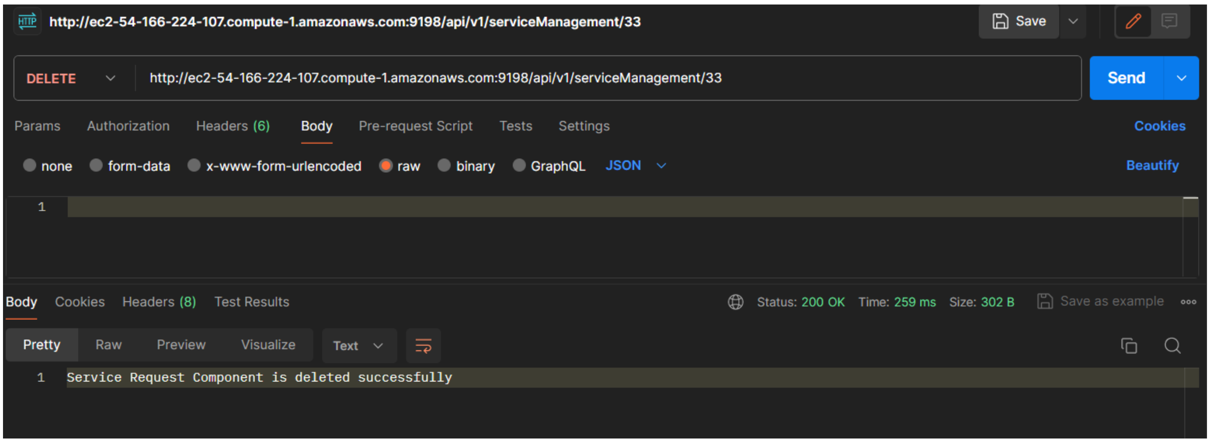
Status: 200 OK Time: 227 ms Size: 5.05 KB

```
[
    {
        "serviceId": 6,
        "projectInfo": "My Project info3",
        "startDate": "2024-01-01",
        "endDate": "2026-01-25",
        "workLocation": "Munich",
        "domain": "Data",
        "role": "Test Engineer",
        "skill": "Unix",
        "technology": "AIML",
        "experience": "Mid level",
        "masterAgreementName": "MAT1"
    },
    {
        "serviceId": 7,
        "projectInfo": "My Project info4",
        "startDate": "2024-01-01",
        "endDate": "2026-01-25",
        "workLocation": "Munich",
        "domain": "Security",
        "role": "Infrastructure Engineer",
        "skill": "Java",
```

Fig 13: API GET

DELETE

The API is available here .The DELETE request serves as a means to remove specific service requests from the table. Users can utilize this operation to withdraw or cancel service requests that are no longer applicable or required. This deletion functionality ensures the flexibility and dynamism of the Service Request table, allowing users to manage their requests effectively.



DELETE http://ec2-54-166-224-107.compute-1.amazonaws.com:9198/api/v1/serviceManagement/33

Status: 200 OK Time: 259 ms Size: 302 B

```
Service Request Component is deleted successfully
```

Fig 14: API DELETE

Deployment

In our web application, the frontend is deployed on AWS, leveraging the robust capabilities of the platform for seamless deployment and management. The AWS infrastructure allows us to automatically create, distribute, and serve our frontend, ensuring high-performance websites and applications. Direct integration with GitHub enables efficient continuous integration, automatic HTTPS, and global Content Delivery Network (CDN) deployment. Whenever updates are made, AWS initiates the build process, rendering markup, and optimizing assets on a dedicated build infrastructure.

For the backend, we have chosen to deploy using Spring Boot for our API and PostgreSQL as our database, both hosted on AWS. This deployment strategy ensures a scalable and reliable backend infrastructure. AWS offers features such as free TLS certificates, a global CDN, DDoS protection, private networks, and seamless auto-deployments from our Git repositories. This comprehensive cloud solution supports the entire spectrum of applications, from straightforward HTML pages to complex systems with multiple microservices. The combination of AWS, Django framework, Spring Boot, and PostgreSQL provides a robust foundation for our web application, offering scalability, security, and efficient management of both frontend and backend components.

Conclusion

In the successful development and deployment of the Service Management Component for the Provider Management Platform, Service Management Component section has played a pivotal role in contributing to the overall project objectives. Our adoption of Agile methodology and the integration of a Kanban board proved instrumental in managing user stories, sprint timelines, and project progress.

Following Agile principles allowed us to respond to changing requirements efficiently. The Kanban board, serving as a visual representation of tasks and their statuses, provided a comprehensive overview of the project landscape. This transparency not only facilitated better communication within our team but also conveyed a clear picture of development progress to stakeholders.

The structured use of the Kanban board for user story assignment enhanced our understanding of tasks, ensuring a cohesive approach to project development. This methodology not only streamlined our workflow but also allowed for effective communication, reducing dependencies on traditional scrum or stand-up meetings.

Ultimately, the Kanban board emerged as a valuable tool, offering visual cues and real-time tracking of progress. The result was improved team communication, enhanced efficiency, and a successful contribution to the development of the Service Management Component within the Provider Management Platform.