JOURNAL LA MULTIAPP

VOL. 05, ISSUE 02 (063-077), 2024
DOI: 10.37899/journallamultiapp.v5i2.1148

Backend Development of a Microservice-Based Website Application for Public Issue Reporting: Case Studyn in People Representative Council

Rizqullah Maziyah Isnaeni[1], Nur Ichsan Utama[1], Sinung Suakanto[1]

[1]Industrial Engineering Faculty, Telkom University, Bandung, Indonesia

**Corresponding Author: Rizqullah Maziyah Isnaeni*
Email: aziyaisnaeni@student.telkomuniversity.ac.id

Article Info

Article history:
Received 23 February 2024
Received in revised form 18 March 2024
Accepted 5 April 2024

Keywords:
Public Issue Reporting
Microservice
Scrum
API Testing
Stress Testing

Abstract

This research introduces a web application aimed at advancing public issue reporting, promoting civic engagement, and expediting government responses. In collaboration with DPRD Jawa Barat, the research leverages Scrum methodology, social network features, and a microservice architecture to create an efficient communication platform between citizens and governmental bodies. The backend of the application, developed using the Go programming language, adopts a microservice architecture to enhance scalability and maintainability. The Scrum methodology facilitates an agile development process, ensuring adaptability to changing requirements and fostering continuous improvement throughout the project lifecycle. Additionally, the study explores the incorporation of social network features to encourage public engagement within the application. This integration allows citizens to connect, share, and discuss public issues, further enhancing the collaborative nature of the reporting platform. To ensure the seamless functionality of the microservices, API testing is employed, validating the reliability and consistency of the application's interfaces. Stress testing is also conducted to assess scalability and performance capabilities, identifying potential optimizations for the system's responsiveness under varying levels of load. In conclusion, this research presents an innovative solution for public issue reporting that combines microservice architecture, Scrum methodology, and social network features. The application's integration of these elements aims to not only streamline citizen-government communication but also create a dynamic platform that encourages active public involvement and collaboration.

Introduction

Public service is a vital aspect of government activities aimed at meeting the needs and rights of the community (Deslatte et al., 2020; Kuziemski & Misuraca, 2020). Within this realm, public complaints serve as a means for individuals to provide feedback, suggestions, criticism, or express concerns regarding the government's performance in delivering public services (Gao et al., 2020; Kim et al., 2022). These complaints, as highlighted by their role in enhancing public service quality and fostering trust in the government, play a crucial role in driving improvements (Ashok et al., 2021). Moreover, public engagement emerges as a powerful tool that not only encourages transparency and accountability but also promotes community participation in public decision-making processes (Denhardt & Denhardt, 2015). By actively engaging citizens, governments can effectively utilize public input to shape policies, prioritize areas for improvement, and ultimately enhance the overall delivery of public services.

ISSN: 2716-3865 (Print), 2721-1290 (Online)

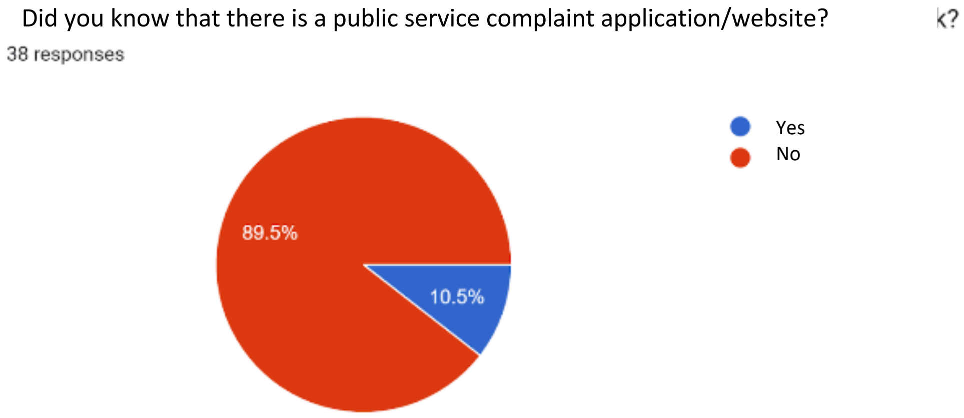


Figure 1. Survey Result of Awareness of Existing Application

After conducting a survey among the residents of Bandung, it was revealed that 89.5% of the respondents were unaware of the existence of any application or website for public issue reporting. This finding highlights a significant gap in knowledge and accessibility, contributing to the inefficiency of submitting complaints effectively and efficiently.

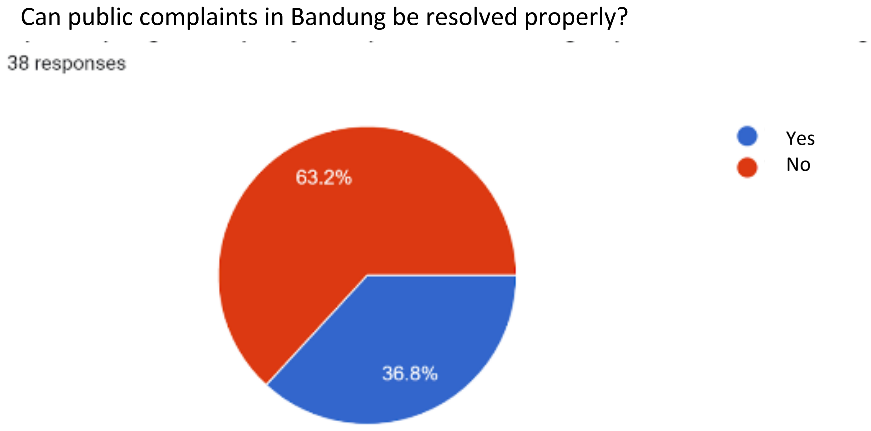


Figure 2. Survey Result of Satisfaction

Furthermore, the survey results indicated that 63.2% of the respondents believed that public issues were not handled effectively. This sentiment underscores the importance of addressing the obstacles and challenges associated with public issue reporting in order to enhance the overall management and resolution of such concerns.

DPRD Jawa Barat was selected as the subject of this case study due to its pivotal role in overseeing the performance of local government and representing the interests of the community within the region. The primary objective of this research is to develop a community complaint website with social network features that effectively addresses the challenges identified through the survey.

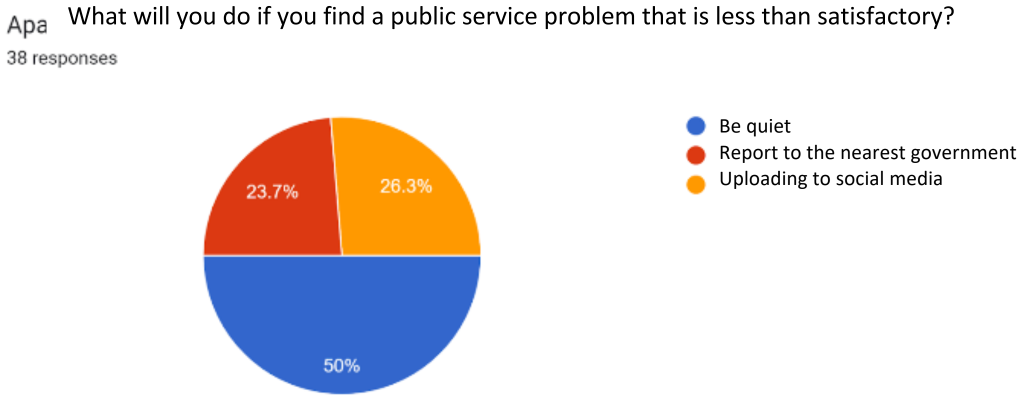


Figure 3. Survey Result of Public Engagement

The survey results revealed that when confronted with a public issue, 50% of the respondents admitted to refraining from taking any action, while 26.3% expressed their intention to disseminate the issue on social media platforms. These findings underscore the significance of the proposed project, which aims to bolster public engagement and mitigate the existing gap in knowledge and accessibility.

By incorporating social networking features into the website, the community will have the opportunity to actively participate, share pertinent information, and engage in discussions related to similar concerns. This social network aspect enhances deeper interaction and connectivity among users, allowing them to offer support, suggest solutions, and collaboratively address the challenges they encounter. Users will be able to view and interact with posts, offering likes and comments to express their feedback and contribute to discussions on the issues they face. This feature mirrors certain elements of social media, allowing users to stay informed about popular posts and engage through likes and comments.

Methods

Scrum

Scrum is an agile software development framework that emphasizes an incremental and iterative approach to managing product development (Sachdeva, 2016; Zayat & Senvar, 2020). This framework challenges traditional, sequential development methods, promoting self-organization within development teams. The collaboration is facilitated through close online interaction and face-to-face communication among team members and various project disciplines.

Application Programming Language (API)

APIs serve as a framework that enables efficient interaction and communication between software applications (Garriga, 2018; Jararweh et al., 2016). They provide guidelines, protocols, and conventions for accessing established software, services, or platforms, allowing developers to integrate functionalities into their applications and leverage existing code and services. APIs act as intermediaries, facilitating seamless collaboration and data exchange between different systems by adhering to common API standards.

A coordinated approach to API adoption is necessary for governments to harness the transformative potential of APIs while mitigating risks associated with loosely-coupled systems (Boyd et al., 2020). By defining API strategies, governments can effectively steer the organizational change management process of digitization efforts.

Stress Testing

Stress testing is a comprehensive assessment technique used to gauge the resilience of a system or entity in the face of extreme events (Plodinec, 2021; Rus et al., 2018). It involves subjecting the system to conditions beyond its typical operational capacity, often pushing it to the breaking point, with the intention of observing the resultant outcomes. In the context of financial literature, stress testing has traditionally focused on evaluating the impact on asset portfolios but has evolved to encompass the analysis of entire banks, banking systems, and financial systems.

In the realm of software development, stress testing is particularly pertinent in validating the performance and durability of applications, networks, or databases. By deliberately imposing conditions that exceed normal operational thresholds, stress testing provides valuable insights into potential bottlenecks, weaknesses, or vulnerabilities that may emerge during periods of heightened demand or unforeseen circumstances. The goal is to uncover how the system behaves under stress, identify its breaking points, and ascertain whether it can gracefully degrade or recover without compromising essential functionalities. This rigorous examination aids developers, engineers, and stakeholders in fortifying the system's infrastructure, enhancing its resilience, and ensuring a robust user experience even in the face of challenging scenarios.

Results and Discussion

In system development, the Scrum method is used to determine development strategies until the system is completed. Then the backend will be tested using API testing and stress testing.

First Iteration

In this phase, functions are determined and planned to be built during the period 16 October – 1 November 2023. During this period a daily scrum is carried out and a sprint review is carried out after the daily scrum ends.

In Table 1 attached is a sprint review containing the backlog that has been built and the review status or description.

Table 1. Sprint Review 1

No	Backlog	Description
1	Login API backend development	As Needed
2	Register API backend development	As Needed
3	Get categories API backend development	As Needed
4	Get commission list API backend development	As Needed
5	Add report API backend development	As Needed
6	Get all report API backend development	As Needed
7	Get report detail API backend development	As Needed
8	Get user list API backend development	As Needed
9	Get user detail API backend development	As Needed

Then, a sprint retrospective is carried out in the form of performance evaluation during the sprint period. Table 2 shows the results of the sprint retrospective on sprint 1.

Tabel 2. Sprint Retrospective 1

Question	Answer
What good things have happened during this sprint?	The sprint went smoothly all backlogs in this sprint were completed on time
What activities didn't work well during this sprint?	Data is not yet integrated with the frontend
What needs to be improved for the next sprint?	Communication between teams must run better

Second Iteration

This phase continues the improvements in sprint 1 and focuses on continuing the API which is the next priority which will be carried out on 2 November – 25 November 2023. Table 3 shows the product backlog in sprint 2 and the results of the review.

Tabel 3. Sprint Review 2

No	Backlog	Description
1	Upload API backend development	As Needed
2	Update complaint API backend development	As Needed
3	Add admin API backend development	As Needed
4	Update admin API backend development	As Needed
5	Get user profile API backend development	As Needed
6	Update profile API backend development	As Needed
7	Get profile photo API backend development	As Needed
8	Like API backend development	As Needed
9	Share API backend development	As Needed

The sprint retrospective can be seen in Table 4.

Table 4. Sprint Retrospective 2

Question	Answer
What good things have happened during this sprint?	The sprint went smoothly All backlogs in this sprint were completed on time Communication between members goes well
What activities didn't work well during this sprint?	The completion of the backlog faces challenges in adjusting data requirements.
What needs to be improved for the next sprint?	It is crucial to ensure better alignment of data needs between the frontend and backend teams.

Third Iteration

In the third sprint, development of the remaining APIs continues. Table 5 contains the product backlog that was worked on in the period 26 November – 31 December 2023. Then a sprint retrospective was carried out which can be seen in Table 6.

Table 5. Sprint Review 3

No	Backlog	Description
1	Add response API backend development	As Needed
2	Add comment API backend development	As Needed
3	Get evidence API backend development	As Needed
4	Forgot password API backend development	As Needed
5	Update password API backend development	As Needed

ISSN: 2716-3865 (Print), 2721-1290 (Online)

6	Create notification API backend development	As Needed
7	Get my notification API backend development	As Needed
8	Verify email API backend development	As Needed
9	Resend verify email API backend	As Needed

Testing

After the development phase of Scrum is complete, the next stage is to test the system using API testing and User stress testing. This aims to ensure that the function can run well and what the condition will be in extreme situations.

API Testing

Tabel 6. API Testing

Description	
Register	**Procedure**
	Access the /register endpoint with the required parameters
	Method
	Post
	Data Example
	Fullname: Lester Mistletoe Whatsapp Number: 081283045710 Email: lestermistletoe@gmail.com Password: testingOnly%2 Confirm_password: testingOnly%2
Login	**Procedure**
	Access the /login endpoint with the required parameters
	Method
	Post
	Data Example
	Email: lestermistletoe@gmail.com Password: testingOnly%2
Get Categories	**Procedure**
	Access the /categories endpoint with the required parameters
	Method
	Get
	Data Example
	-
Get Commission List	**Procedure**
	Access the /list-komisi endpoint with the required parameters
	Method
	Get
	Data Example
	-
Add Report	**Procedure**
	Access the /complaint endpoint with the required parameters
	Method
	Post
	Data Example
	Title: Penggalian PDAM membuat macet Description: Testing Location: 222

	Date: 2023-12-09T05:00:27+07:00 Category ID: 11 is_anonymous: false
Get All Reports	**Procedure**
	Access the /complaint endpoint with the required parameters
	Method
	Get
	Data Example
	-
Get Report Detail	**Procedure**
	Access the /complaint/:id endpoint with the required parameters
	Method
	Get
	Data Example
	Id: 7
Get User List	**Procedure**
	Access the /user/list-user endpoint with the required parameters
	Method
	Get
	Data Example
	-
Get User Detail	**Procedure**
	Access the /user/profile endpoint with the required parameters
	Method
	Get
	Data Example
	Id: 22
Upload	**Procedure**
	Access the /storage/upload endpoint with the required parameters
	Method
	Post
	Data Example
	File: cosmos.jpg Complaint_id: 7
Update Complaint	**Procedure**
	Access the /complaint endpoint with the required parameters
	Method
	Put
	Data Example
	Complaint_id: 7 Category_id: 5 Status: Dalam Proses
Add Admin	**Procedure**
	Access the /admin endpoint with the required parameters
	Method
	Post
	Data Example
	Fullname: Admin DPRD Subang Email: admin_dprd_subang@yopmail.com Role_id: 4

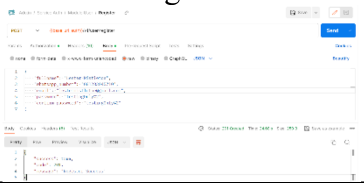
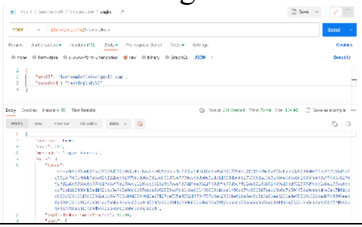
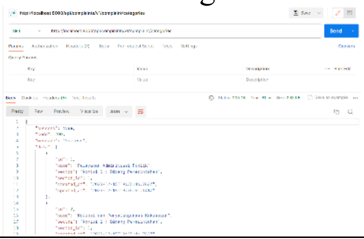
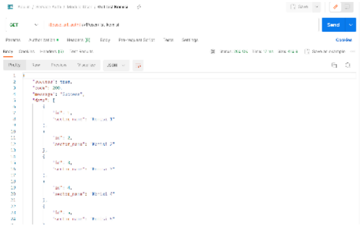
	Status: Active Sector_id: 3 Profile_photo: https://img.freepik.com/premium-vector/man-avatar-profile-picture-vector-illustration_268834-538.jpg Password: Aduin#2023 Confirm_password: Aduin#2023
Update Admin	**Procedure**
	Access the /admin endpoint with the required parameters
	Method
	Put
	Data Example
	User_id: 22 Fullname: Admin DPRD Tangerang Profile_photo: https://img.freepik.com/premium-vector/man-avatar-profile-picture-vector-illustration_268834-538.jpg Whatsapp_number: 081234567890 Address: Jl. Cerah Abadi Sector_id: 2 Status: Active
Get USer Profile	**Procedure**
	Access the /profile endpoint with the required parameters
	Method
	Get
	Data Example
	-
Update Profile	**Procedure**
	Access the /profile endpoint with the required parameters
	Method
	Put
	Data Example
	Fullname: Rizqullah Maziyah Isnaeni, Profile_photo: https://img.freepik.com/premium-vector/man-avatar-profile-picture-vector-illustration_268834-538.jpg, Whatsapp_number: 081283041234, Address: Jl. Menuju 1001 Kebahagiaan Yang Haqiqi
Get Profile Photo	**Procedure**
	Access the storage/profile endpoint with the required parameters
	Method
	Get
	Data Example
	Filename: XWareDALbmAt.jpg
Like	**Procedure**
	Access the /complaint/like endpoint with the required parameters
	Method
	Put
	Data Example
	Id: 7
Share	**Procedure**
	Access the /complaint/share endpoint with the required parameters
	Method
	Put

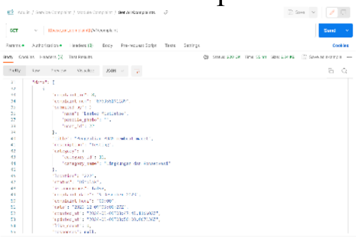
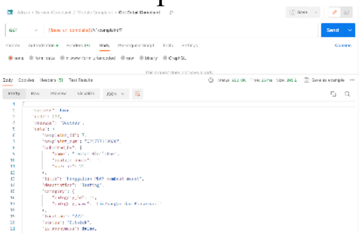
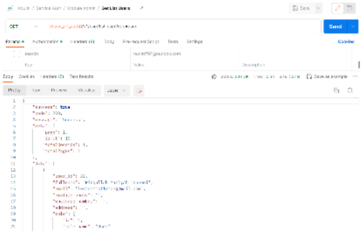
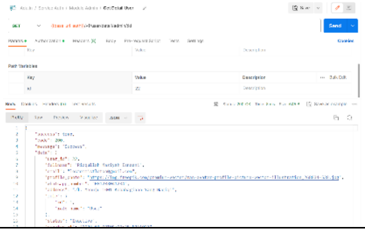
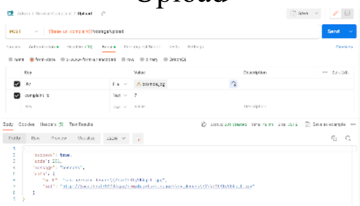
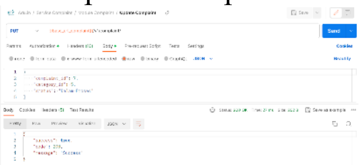
ISSN: 2716-3865 (Print), 2721-1290 (Online)

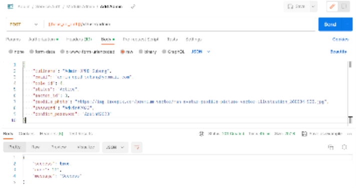
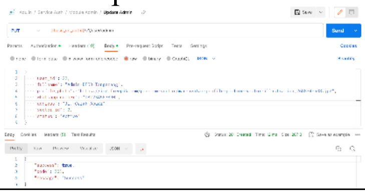
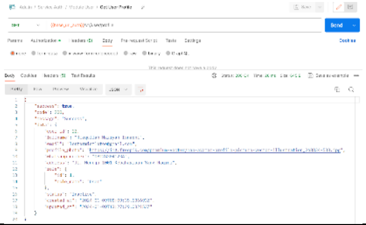
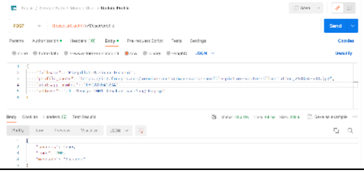
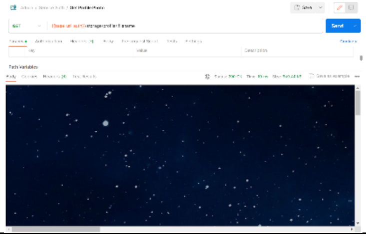
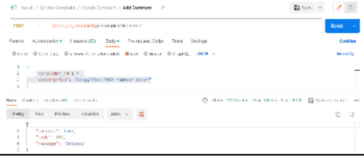
	Data Example
	Id: 7
Add Response	**Procedure**
	Access the /response endpoint with the required parameters
	Method
	Post
	Data Example
	Complaint_id: 4 Description: Akan segera ditindaklanjuti oleh dinas terkait
Add Comment	**Procedure**
	Access the /complaint/comment endpoint with the required parameters
	Method
	Post
	Data Example
	Complaint_id: 7, Description: Penggalian PDAM membuat macet
Get Evidence	**Procedure**
	Access the /storage/evidences/:complaint_id/:location endpoint with the required parameters
	Method
	Get
	Data Example
	Complaint_id: 7 Location: dQtKfyVBXjLI.jpg
Forgot Password	**Procedure**
	Access the /forgot-password endpoint with the required parameters
	Method
	Post
	Data Example
	Email: lestermistletoe@gmail.com
Update Password	**Procedure**
	Access the /update-password endpoint with the required parameters
	Method
	Put
	Data Example
	Password: aduin1234 Confirm_password: aduin1234
Create Notification	**Procedure**
	Access the /notification endpoint with the required parameters
	Method
	Post
	Data Example
	Title: Pemerintah Kabupaten User_id: 22 Description: Pemerintah Kabupaten memberi tindakan <b>Awas Tukang Bakso Lewat</b> Icon: https://pinrangkab.go.id/wp-content/uploads/2019/07/LOGO-KABUPATEN-PINRANG-263x300-263x300.png
Get My Notification	**Procedure**
	Access the /notification endpoint with the required parameters

	Method
	Get
	Data Example
	-
Verify Email	**Procedure**
	Access the /verify-email endpoint with the required parameters
	Method
	Post
	Data Example
	Token send from mail
Resend Verify Mail	**Procedure**
	Access the /resend-verify-email endpoint with the required parameters
	Method
	Post
	Data Example
	Email: azalea.eon@gmail.com

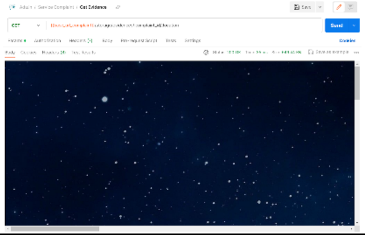
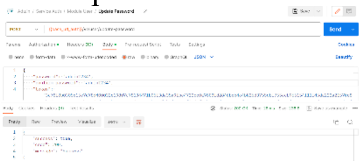
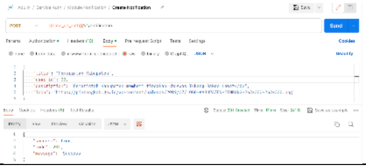
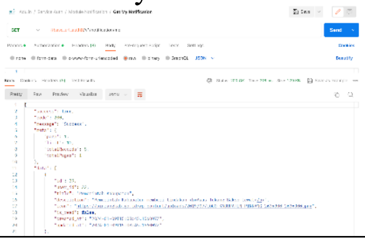
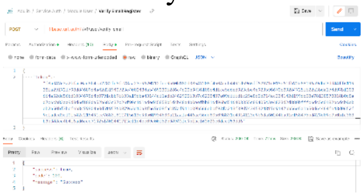
Tabel 7. API Testing Result

No	**API Output**	**Result**	
		Success	**Fail**
1	Register 	✓	
2	Login 	✓	
3	Get Categories 	✓	
4	Get Commission List 	✓	
5	Add Report	✓	

6	Get All Reports 	✓	
7	Get Report Detail 	✓	
8	Get User List 	✓	
9	Get User Detail 	✓	
10	Upload 	✓	
11	Update Complaint 	✓	
12	Add Admin	✓	

			
13	Update Admin 	√	
14	Get User Profile 	√	
15	Update Profile 	√	
16	Get Profile Photo 	√	
17	Like 	√	
18	Share 	√	
19	Add Response 	√	
20	Add Comment 	√	

ISSN: 2716-3865 (Print), 2721-1290 (Online)

21	Get Evidence 	√	
22	Forgot Password 	√	
23	Update Password 	√	
24	Create Notification 	√	
25	Get My Notification 	√	
26	Verify Email 	√	
27	Resend Verify Mail	√	

Stress Testing

The stress testing focused on login authorization and data retrieval due to their critical roles in user interaction. Evaluating login functionality ensures the system's security and stability under a surge of simultaneous logins. Testing data retrieval assesses the system's capacity to serve multiple users concurrently, providing insights into performance and highlighting areas for improvement. The timeframe of each scenario is 10 seconds.

Tabel 8. Login Authentication Results

Timeframe	User	Completed in Seconds
10	100	9
10	300	10
10	400	12

Tabel 9. Retrieval Data Results

Timeframe	User	Completed in Seconds
10	100	10
10	200	14
10	300	19

Conclusion

The backend has functioned effectively, with all data within the Adu.In website being successfully stored in the database. The accessed data can be displayed separately according to the roles of the respective actors. Developing the backend using the Scrum methodology has significantly impacted the effectiveness of application development. With the Scrum methodology, the feature development process can be more focused, thanks to the presence of a backlog. By implementing the microservices architecture, the backend is able to operate optimally by utilizing asynchronous APIs. This asynchronous approach allows for efficient and effective communication processes among backend services. Therefore, the microservices architecture has demonstrated its ability to optimize backend performance through the use of asynchronous APIs, enhancing overall system scalability and responsiveness.

Following API testing, it was found that the functions within the Adu.In website operate seamlessly. Additionally, stress testing revealed that as the number of users attempting to access Adu.In increases, the loading process slows down. Within 10 seconds, Adu.In can handle the login process for 300 users and data retrieval processes for 100 users.

References

Ashok, M., Al Badi Al Dhaheri, M. S. M., Madan, R., & Dzandu, M. D. (2021). How to counter organisational inertia to enable knowledge management practices adoption in public sector organisations. *Journal of Knowledge Management, 25*(9), 2245–2273. https://doi.org/10.1108/JKM-09-2020-0700

Boyd, M., Vaccari, L., Posada, M., & Gattwinkel, D. (2020). An Application Programming Interface (API) framework for digital government. *European Commission, Joint Research Centre*.

Denhardt, J. V, & Denhardt, R. B. (2015). The new public service revisited. *Public Administration Review, 75*(5), 664–672.

Deslatte, A., Hatch, M. E., & Stokan, E. (2020). How can local governments address pandemic inequities? *Public Administration Review, 80*(5), 827–831. https://doi.org/10.1111/puar.13257

Gao, Z., Wang, S., & Gu, J. (2020). Public participation in smart-city governance: a qualitative content analysis of public comments in urban China. *Sustainability, 12*(20), 8605. https://doi.org/10.3390/su12208605

Garriga, M. (2018). Towards a taxonomy of microservices architectures. In *Software Engineering and Formal Methods: SEFM 2017 Collocated Workshops: DataMod, FAACS, MSE, CoSim-CPS, and FOCLASA, Trento, Italy, September 4-5, 2017, Revised Selected Papers 15* (pp. 203-218). Springer International Publishing.

Jararweh, Y., Doulat, A., Darabseh, A., Alsmirat, M., Al-Ayyoub, M., & Benkhelifa, E. (2016, April). SDMEC: Software defined system for mobile edge computing. In *2016 IEEE international conference on cloud engineering workshop (IC2EW)* (pp. 88-93). IEEE. https://doi.org/10.1109/IC2EW.2016.45

Kim, N. Y., Kim, H. J., & Kim, S. H. (2022). Are satisfied citizens willing to participate more? An analysis of citizens' life satisfaction in the public service domain and public participation. *International Review of Public Administration*, *27*(3), 211-227. https://doi.org/10.1080/12294659.2022.2119650

Plodinec, M. J. (2021). Stress Testing to Assess Recovery from Extreme Events. *Journal of Homeland Security and Emergency Management, 18*(2), 151–176. https://doi.org/10.1515/jhsem-2020-0012

Rus, K., Kilar, V., & Koren, D. (2018). Resilience assessment of complex urban systems to natural disasters: A new literature review. *International Journal of Disaster Risk Reduction, 31*, 311–330. https://doi.org/10.1016/j.ijdrr.2018.05.015

Sachdeva, S. (2016). Agile Methodologies. *International Journal of Computer Science and Information Technologies, 7*(2), 40–44.

Zayat, W., & Senvar, O. (2020). Framework study for agile software development via scrum and Kanban. *International Journal of Innovation and Technology Management, 17*(04), 2030002. https://doi.org/10.1142/S0219877020300025

ISSN: 2716-3865 (Print), 2721-1290 (Online)