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

Rizqullah Maziyah Isnaeni¹, Nur Ichsan Utama¹, Sinung Suakanto¹

¹Industrial Engineering Faculty, Telkom University, Bandung, Indonesia

*Corresponding Author: Rizqullah Maziyah Isnaeni
Email: aziva_isnaeni@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.
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.
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>
<thead>
<tr>
<th>No</th>
<th>Backlog</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Login API backend development</td>
<td>As Needed</td>
</tr>
<tr>
<td>2</td>
<td>Register API backend development</td>
<td>As Needed</td>
</tr>
<tr>
<td>3</td>
<td>Get categories API backend development</td>
<td>As Needed</td>
</tr>
<tr>
<td>4</td>
<td>Get commission list API backend development</td>
<td>As Needed</td>
</tr>
<tr>
<td>5</td>
<td>Add report API backend development</td>
<td>As Needed</td>
</tr>
<tr>
<td>6</td>
<td>Get all report API backend development</td>
<td>As Needed</td>
</tr>
<tr>
<td>7</td>
<td>Get report detail API backend development</td>
<td>As Needed</td>
</tr>
<tr>
<td>8</td>
<td>Get user list API backend development</td>
<td>As Needed</td>
</tr>
<tr>
<td>9</td>
<td>Get user detail API backend development</td>
<td>As Needed</td>
</tr>
</tbody>
</table>

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

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

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

<table>
<thead>
<tr>
<th>No</th>
<th>Backlog</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Upload API backend development</td>
<td>As Needed</td>
</tr>
<tr>
<td>2</td>
<td>Update complaint API backend development</td>
<td>As Needed</td>
</tr>
<tr>
<td>3</td>
<td>Add admin API backend development</td>
<td>As Needed</td>
</tr>
<tr>
<td>4</td>
<td>Update admin API backend development</td>
<td>As Needed</td>
</tr>
<tr>
<td>5</td>
<td>Get user profile API backend development</td>
<td>As Needed</td>
</tr>
<tr>
<td>6</td>
<td>Update profile API backend development</td>
<td>As Needed</td>
</tr>
<tr>
<td>7</td>
<td>Get profile photo API backend development</td>
<td>As Needed</td>
</tr>
<tr>
<td>8</td>
<td>Like API backend development</td>
<td>As Needed</td>
</tr>
<tr>
<td>9</td>
<td>Share API backend development</td>
<td>As Needed</td>
</tr>
</tbody>
</table>

The sprint retrospective can be seen in Table 4.

Table 4. Sprint Retrospective 2

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

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

<table>
<thead>
<tr>
<th>No</th>
<th>Backlog</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Add response API backend development</td>
<td>As Needed</td>
</tr>
<tr>
<td>2</td>
<td>Add comment API backend development</td>
<td>As Needed</td>
</tr>
<tr>
<td>3</td>
<td>Get evidence API backend development</td>
<td>As Needed</td>
</tr>
<tr>
<td>4</td>
<td>Forgot password API backend development</td>
<td>As Needed</td>
</tr>
<tr>
<td>5</td>
<td>Update password API backend development</td>
<td>As Needed</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Description</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register</td>
<td>Access the /register endpoint with the required parameters</td>
</tr>
<tr>
<td></td>
<td>Method Post</td>
</tr>
<tr>
<td></td>
<td>Data Example</td>
</tr>
<tr>
<td></td>
<td>Fullname: Lester Mistletoe</td>
</tr>
<tr>
<td></td>
<td>Whatsapp Number: 081283045710</td>
</tr>
<tr>
<td></td>
<td>Email: <a href="mailto:lestermistletoe@gmail.com">lestermistletoe@gmail.com</a></td>
</tr>
<tr>
<td></td>
<td>Password: testingOnly%2</td>
</tr>
<tr>
<td></td>
<td>Confirm_password: testingOnly%2</td>
</tr>
<tr>
<td>Login</td>
<td>Access the /login endpoint with the required parameters</td>
</tr>
<tr>
<td></td>
<td>Method Post</td>
</tr>
<tr>
<td></td>
<td>Data Example</td>
</tr>
<tr>
<td></td>
<td>Email: <a href="mailto:lestermistletoe@gmail.com">lestermistletoe@gmail.com</a></td>
</tr>
<tr>
<td></td>
<td>Password: testingOnly%2</td>
</tr>
<tr>
<td>Get Categories</td>
<td>Access the /categories endpoint with the required parameters</td>
</tr>
<tr>
<td></td>
<td>Method Get</td>
</tr>
<tr>
<td></td>
<td>Data Example</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Get Commission List</td>
<td>Access the /list-komisi endpoint with the required parameters</td>
</tr>
<tr>
<td></td>
<td>Method Get</td>
</tr>
<tr>
<td></td>
<td>Data Example</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Add Report</td>
<td>Access the /complaint endpoint with the required parameters</td>
</tr>
<tr>
<td></td>
<td>Method Post</td>
</tr>
<tr>
<td></td>
<td>Data Example</td>
</tr>
<tr>
<td></td>
<td>Title: Penggalian PDAM membuat macet</td>
</tr>
<tr>
<td></td>
<td>Description: Testing</td>
</tr>
<tr>
<td></td>
<td>Location: 222</td>
</tr>
</tbody>
</table>

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.
<table>
<thead>
<tr>
<th>Category ID: 11</th>
<th>is_anonymous: false</th>
</tr>
</thead>
</table>

### 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**

- Id: 22

### 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
| **Update Admin** | **Status:** Active  
**Sector_id:** 3  
**Password:** Aduin#2023  
**Confirm_password:** Aduin#2023 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Procedure</strong></td>
<td><strong>Access the /admin endpoint with the required parameters</strong></td>
</tr>
<tr>
<td><strong>Method</strong></td>
<td><strong>Put</strong></td>
</tr>
</tbody>
</table>
| **Data Example** | **User_id:** 22  
**Fullname:** Admin DPRD Tangerang  
**Whatsapp_number:** 081234567890  
**Address:** Jl. Cerah Abadi  
**Sector_id:** 2  
**Status:** Active |
| **Get User Profile** | **Access the /profile endpoint with the required parameters** |
| **Method** | **Get** |
| **Data Example** | - |
| **Update Profile** | **Access the /profile endpoint with the required parameters** |
| **Method** | **Put** |
| **Data Example** | **Fullname:** Rizquillah 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 |
<p>| <strong>Get Profile Photo</strong> | <strong>Access the storage/profile endpoint with the required parameters</strong> |
| <strong>Method</strong> | <strong>Get</strong> |
| <strong>Data Example</strong> | <strong>Filename:</strong> XWareDALbmAt.jpg |
| <strong>Like</strong> | <strong>Access the /complaint/like endpoint with the required parameters</strong> |
| <strong>Method</strong> | <strong>Put</strong> |
| <strong>Data Example</strong> | <strong>Id:</strong> 7 |
| <strong>Share</strong> | <strong>Access the /complaint/share endpoint with the required parameters</strong> |
| <strong>Method</strong> | <strong>Put</strong> |</p>
<table>
<thead>
<tr>
<th>Component</th>
<th>Data Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Response</td>
<td><strong>Procedure</strong>&lt;br&gt;Access the /response endpoint with the required parameters&lt;br&gt;<strong>Method</strong> Post&lt;br&gt;<strong>Data Example</strong>&lt;br&gt;Complaint_id: 4&lt;br&gt;Description: Akan segera ditindaklanjuti oleh dinas terkait</td>
</tr>
<tr>
<td>Add Comment</td>
<td><strong>Procedure</strong>&lt;br&gt;Access the /complaint/comment endpoint with the required parameters&lt;br&gt;<strong>Method</strong> Post&lt;br&gt;<strong>Data Example</strong>&lt;br&gt;Complaint_id: 7&lt;br&gt;Description: Penggalian PDAM membuat macet</td>
</tr>
<tr>
<td>Get Evidence</td>
<td><strong>Procedure</strong>&lt;br&gt;Access the /storage/evidences/:complaint_id/:location endpoint with the required parameters&lt;br&gt;<strong>Method</strong> Get&lt;br&gt;<strong>Data Example</strong>&lt;br&gt;Complaint_id: 7&lt;br&gt;Location: dQtKfyVBxjLI.jpg</td>
</tr>
<tr>
<td>Forgot Password</td>
<td><strong>Procedure</strong>&lt;br&gt;Access the /forgot-password endpoint with the required parameters&lt;br&gt;<strong>Method</strong> Post&lt;br&gt;<strong>Data Example</strong>&lt;br&gt;Email: <a href="mailto:lestermistletoe@gmail.com">lestermistletoe@gmail.com</a></td>
</tr>
<tr>
<td>Update Password</td>
<td><strong>Procedure</strong>&lt;br&gt;Access the /update-password endpoint with the required parameters&lt;br&gt;<strong>Method</strong> Put&lt;br&gt;<strong>Data Example</strong>&lt;br&gt;Password: aduin1234&lt;br&gt;Confirm_password: aduin1234</td>
</tr>
<tr>
<td>Create Notification</td>
<td><strong>Procedure</strong>&lt;br&gt;Access the /notification endpoint with the required parameters&lt;br&gt;<strong>Method</strong> Post&lt;br&gt;<strong>Data Example</strong>&lt;br&gt;Title: Pemerintah Kabupaten&lt;br&gt;User_id: 22&lt;br&gt;Description: Pemerintah Kabupaten memberi tindakan &lt;b&gt;Awas Tukang Bakso Lewat&lt;/b&gt;&lt;br&gt;Icon: <a href="https://pinrangkab.go.id/wp-content/uploads/2019/07/LOGO-KABUPATEN-PINRANG-263x300.png">https://pinrangkab.go.id/wp-content/uploads/2019/07/LOGO-KABUPATEN-PINRANG-263x300.png</a></td>
</tr>
<tr>
<td>Get My Notification</td>
<td><strong>Procedure</strong>&lt;br&gt;Access the /notification endpoint with the required parameters</td>
</tr>
<tr>
<td>No</td>
<td>API Output</td>
</tr>
<tr>
<td>----</td>
<td>-------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Register</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Login</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Get Categories</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Get Commission List</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Add Report</td>
</tr>
</tbody>
</table>

**Procedure**

- Access the /verify-email endpoint with the required parameters

**Data Example**

- Token send from mail

- Email: azalea.eon@gmail.com

**Table 7. API Testing Result**
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Get All Reports</td>
<td>✓</td>
</tr>
<tr>
<td>7</td>
<td>Get Report Detail</td>
<td>✓</td>
</tr>
<tr>
<td>8</td>
<td>Get User List</td>
<td>✓</td>
</tr>
<tr>
<td>9</td>
<td>Get User Detail</td>
<td>✓</td>
</tr>
<tr>
<td>10</td>
<td>Upload</td>
<td>✓</td>
</tr>
<tr>
<td>11</td>
<td>Update Complaint</td>
<td>✓</td>
</tr>
<tr>
<td>12</td>
<td>Add Admin</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>13</td>
<td>Update Admin</td>
<td>✓</td>
</tr>
<tr>
<td>14</td>
<td>Get User Profile</td>
<td>✓</td>
</tr>
<tr>
<td>15</td>
<td>Update Profile</td>
<td>✓</td>
</tr>
<tr>
<td>16</td>
<td>Get Profile Photo</td>
<td>✓</td>
</tr>
<tr>
<td>17</td>
<td>Like</td>
<td>✓</td>
</tr>
<tr>
<td>18</td>
<td>Share</td>
<td>✓</td>
</tr>
<tr>
<td>19</td>
<td>Add Response</td>
<td>✓</td>
</tr>
<tr>
<td>20</td>
<td>Add Comment</td>
<td>✓</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Timeframe</th>
<th>User</th>
<th>Completed in Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>100</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>300</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>400</td>
<td>12</td>
</tr>
</tbody>
</table>
Tabel 9. Retrieval Data Results

<table>
<thead>
<tr>
<th>Timeframe</th>
<th>User</th>
<th>Completed in Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>100</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>200</td>
<td>14</td>
</tr>
<tr>
<td>10</td>
<td>300</td>
<td>19</td>
</tr>
</tbody>
</table>

**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**


