Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Implementation of PWA in Scholarship Application Using Microservices Architecture for Enhancing User Engagement

Sri Andriati Asri[1] ✉ I Gusti Ngurah Bagus Caturbawa[2], Putu Manik Prihatini[3], Ni Wayan Rasmini[4], Made Riyan Adi Nugroho[5], Elina Rudiastari[6] and Widyadi Setiawan[7]

[12356]*Information Technology Department, Politeknik Negeri Bali, Badung, Indonesia*
[4]*Electrical Engineering Department, Politeknik Negeri Bali, Badung, Indonesia*
[7]*Electrical Engineering Department, Udayana University, Badung, Indonesia*
Corresponding Author: Sri Andriati Asri, **E-mail**: sriandriati@pnb.ac.id

| ABSTRACT

The shortage of traditional web development makes researchers turn to web development using Progressive Web App (PWA) in developing a web, which has been increasingly done. PWA has the ability to function even with a low internet connection, maintain interactive user and access speeds, and provide real-time notifications. In modern web development, where the demand for data, features, and users can increase rapidly, an architecture is required to handle this problem. Microservices architecture can answer this. This research will develop a scholarship application using PWA and microservices architecture. The scholarship application was developed using Veu.js for the front end of the client application, MySQL for the application database, and the SLIM framework for microservices implementation. Furthermore, we utilized RAbbitMQ to develop a message broker and Docker for microservices contaner. The scholarship application developed using PWA technology and microservices in this research has been tested by conducting a User Acceptance Test. The results are the implementation of PWA in the scholarship application using microservice architecture has worked well, and the application performance has been able to overcome problems in terms of scalability, speed, and reliability.

| KEYWORDS

Scholarship application, Progressive Web App, Microservices.

| ARTICLE INFORMATION

ACCEPTED: 01 October 2024 **PUBLISHED:** 27 October 2024 **DOI:** 10.32996/jcsts.2024.6.5.2

1. Introduction

In recent years, traditional web development using monolithic architecture has been obliterated. Researchers have started to switch to developing the web in a more agile and flexible manner. In addition to having limited interaction with users in real time, the traditional web also has problems in terms of scalability. Furthermore, the traditional web is highly dependent on the internet connection's quality. This causes applications developed with traditional and monolithic technologies to be difficult to access in locations with low internet connection quality. Furthermore, these systems do not support the real-time notification feature, which is critical for informing users about important updates such as the app's status. Scalability and flexibility can be hampered by the traditional monolithic structure (Smyk et al., 2023). In addition, conventional web architecture often creates trust issues among stakeholders because it does not provide the transparency required for effective scholarship management (Swati & Nitin, n.d.).

Progressive Web Apps (PWA) have emerged as a favorable solution to overcome the limitations of conventional web applications. PWAs allow apps to run offline, provide push notifications, and are responsive across multiple devices, enhancing the user experience (Adetunji et al., 2020), (Adetunji et al., 2023). In contrast, microservice architectures are gaining popularity due to their ability to overcome the limitations of monolithic architectures. Microservices break down applications into small, independent services, which allows for better scalability, greater development flexibility, and ease of application maintenance. Although this technology has many advantages, more research needs to be done to find out how it affects user engagement. Therefore, it is important to see how combining PWAs with microservices architecture can improve application performance and scalability as well as increase user engagement. In the end, this research created a scholarship application with PWA and used microservice architecture to improve the application's performance and scalability.

2. Literature Review

Contemporary web development technologies such as Progressive Web Apps (PWA) have brought about major changes in web application development. Many researchers have previously investigated the ability of PWAs to enhance user experience with offline features and high performance on various devices. The results of previous studies show that the implementation of PWAs improves user experience (Pikuliak et al., n.d.), (Adetunji et al., 2023), (Yeh et al., 2019). Additionally, PWAs excel over traditional web apps in terms of user interaction and engagement as they support improved accessibility, SEO, and ease of installation directly from the browser (Nugraha et al., 2022). Research has also been carried out by combining PWA and Firebase to create push notifications without opening the app or software, as long as the browser is active and connected to the internet (Dwi Purnomo Putro et al., 2023).

In web development, microservice architecture has been used to address scalability issues and ease of handling application errors (Blinowski et al., 2022). Microservice architecture divides the application into separate and independent service modules. Each service module has a specific task that can be performed separately and independently (Irudayaraj, 2019). According to additional research on microservices deployment, microservices improve web development by enabling a scalable, modular architecture that supports continuous automated deployment and scaling, optimizes resource usage and improves system resilience (Bravetti et al., 2019). In his research, Waseem discovered that microservices architecture allows web applications to be scalable and maintainable, enabling domain-based design, API management, and effective monitoring and testing practices, as well as making them independent services (Waseem et al., 2021). Bravetti also said that microservices architecture helps web development because it enables a scalable, modular architecture that supports continuous automated deployment and scaling, optimizes resource usage, and improves system resilience with loosely coupled services (Bravetti et al., 2019). Additionally, they conducted research on the implementation of microservices (Asri et al., 2022). Due to the complexity of processes and data as well as the large number of potential users, microservices architecture is used to address scalability and quick response application issues when developing smart village applications.

3. Methodology

This research method will be described in several stages, from literature review, planning and user requirement, design, implementation, system integration, user testing, and analysis. Figure 1 shows the research methodology.

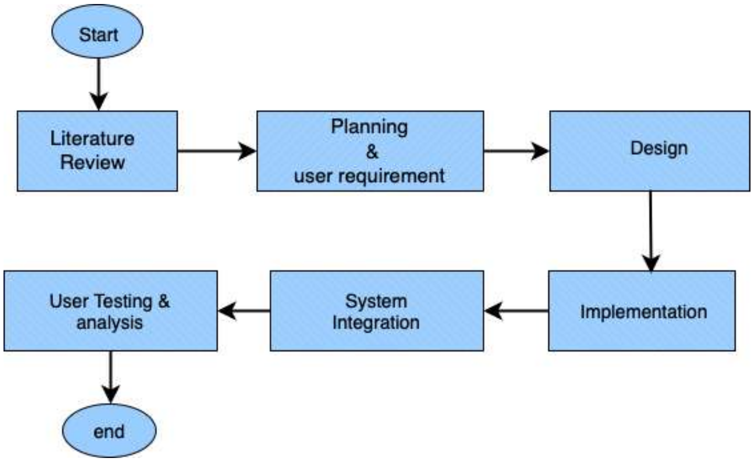


Figure 1 Research methodology

3.1 Literatur review

This stage is the stage of searching and studying research on PWA and microservices that have been carried out by other researchers. We are examining the benefits and drawbacks, along with the potential for growth through the implementation of PWA and microservices technology. Additionally, the researcher will examine how PWAs can enhance application performance and user engagement and how microservices can boost system flexibility and scalability. The researcher compiles the results once the analyses are complete. The results highlight the differences from previous research, particularly when it comes to the application of the two technologies in scholarships. Additionally, suggestions have been made to address the gaps in this research. Many researchers have developed web applications or PWAs to address issues like low internet connection, user engagement, and push notifications during inactivity. Many researchers also use the microservices architecture to divide applications into various modular services, thereby addressing scalability issues and enhancing development and maintenance flexibility. This study aims to create a scholarship web that integrates PWA and microservices architecture, as the integration of these two elements in application development remains relatively unexplored.

3.2 Planning and user requirement

We have established the functional and non-functional requirements for the scholarship application at this stage. We conducted surveys and interviews with stakeholders, including the scholarship administration and students, to determine user requirements. The survey results determined the functional and non-functional requirements. Next, we conducted an analysis to assess the PWA and microservices, which included service modularization, push notifications, and offline features.

3.3 Design

The design process includes several important components, such as user interface design, database design, user interaction design, and system architecture design. Database design is to develop an efficient data storage structure by considering inter-entity relationships, data normalisation, and query optimisation. To display the interaction of users with the system, a usecase diagram is made. Finally, the system architecture design determines how the main components of the system interact and are organised as a whole. At this stage, the designer determines whether the system will use a monolithic or microservices architecture, how the system will handle external components such as APIs or third-party services, and the communication mechanism between services.

3.4 Implementation and system integration

This stage is where the design is converted into coding using a programming language that supports PWA, including the application database. Furthermore, the application features that are built will be integrated into the scholarship application. The system functionality will be tested using the blackblock testing method.

3.5 User testing and analysis

The application's performance will be tested by users, such as students and admins. The scenario is that users will be given a username and password to log in to the application and then access the features of the application. Furthermore, users will be asked to fill out a questionnaire about the performance of the application.

4. Results and Discussion

4.1 User requirement results

After the requirement process was carried out, the student data to be used was taken from student data from the academic system. The functional and non-functional requirements are defined. Table 1 shows the functional and non-functional requirements.

Table 1 The functional and non-functional requirements

Functional Requirements	Non-functional Requirements
Profile	Security
Submission	Performance
Notifications	Scalability
Selection	Availability
Search	Cross-platform Compatibility
Status Tracking	Usability
Add schemes	Maintainability
Report	Technology

The technology refers to the programming languages and hardware needed to develop the scholarship application. Table 2 shows the programming language technology that was used.

Table 2 Programming language to develop scholarship application

Programming Language Technology	Description
Vue.js	Front-end framework to create client application
SLIM Framework	PHP micro-framework to create microservices
MySQL	Database application
RabbitMQ	An open source to create message broker
Docker	Microservices container

4.2 Design results

The design stage results in such as use case diagrams, database, and system architecture. Figure 2 shows the use case diagram of the application. Admins are employees assigned to manage scholarships, whereas students are students who are eligible to apply for scholarships.

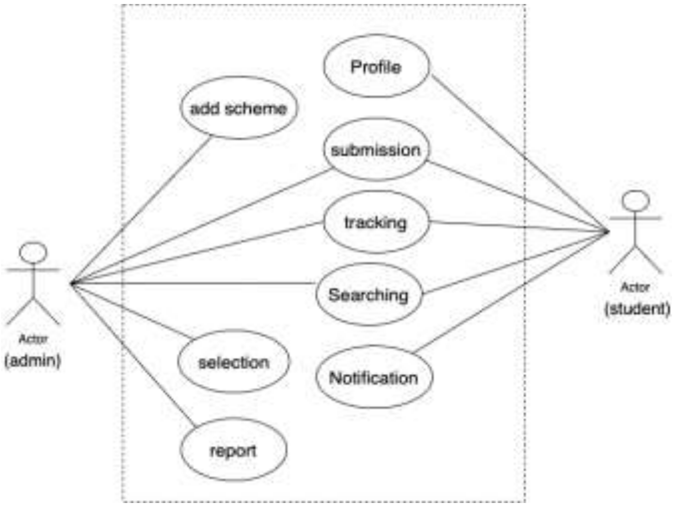
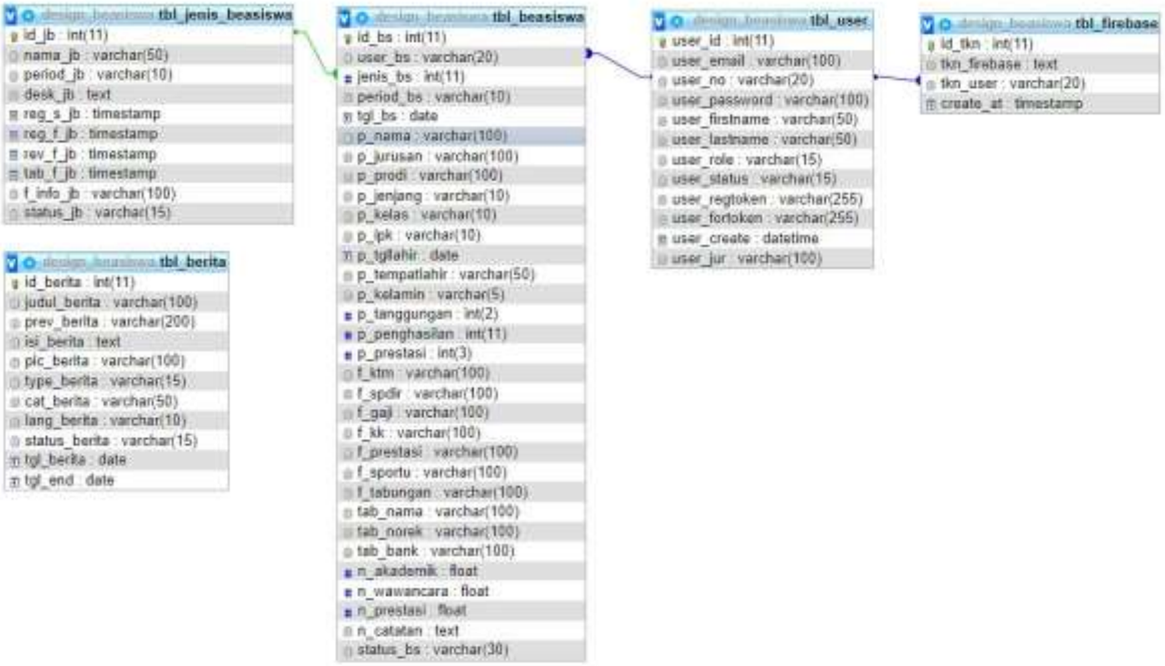


Figure 2 Usecase diagram scholarship application

Figure 3 shows the database of scholarship applications. The database is created using MySQL and consists of 5 tables, which are table jenis_beasiswa (scholarship scheme), table beasiswa (scholarship), table user, table berita (news), and table firebase.

The system architecture used microservices architecture, which consists of several separate services that are functional independently. The system architecture is shown in Figure 4.



design_beasiswa tbl_jenis_beasiswa
id_jb : int(11)
nama_jb : varchar(50)
period_jb : varchar(10)
desk_jb : text
reg_s_jb : timestamp
reg_f_jb : timestamp
rev_f_jb : timestamp
tab_f_jb : timestamp
f_info_jb : varchar(100)
status_jb : varchar(15)

design_beasiswa tbl_berita
id_berita : int(11)
judul_berita : varchar(100)
prev_berita : varchar(200)
isi_berita : text
pic_berita : varchar(100)
type_berita : varchar(15)
cat_berita : varchar(50)
lang_berita : varchar(10)
status_berita : varchar(15)
tgl_berita : date
tgl_end : date

design_beasiswa tbl_beasiswa
id_bs : int(11)
user_bs : varchar(20)
jenis_bs : int(11)
period_bs : varchar(10)
tgl_bs : date
p_nama : varchar(100)
p_jurusan : varchar(100)
p_prodi : varchar(100)
p_jenjang : varchar(10)
p_kelas : varchar(10)
p_ipk : varchar(10)
p_tgllahir : date
p_tempatlahir : varchar(50)
p_kelamin : varchar(5)
p_tanggungan : int(2)
p_penghasilan : int(11)
p_prestasi : int(3)
f_ktm : varchar(100)
f_spdir : varchar(100)
f_gaji : varchar(100)
f_kk : varchar(100)
f_prestasi : varchar(100)
f_sportu : varchar(100)
f_tabungan : varchar(100)
tab_nama : varchar(100)
tab_norek : varchar(100)
tab_bank : varchar(100)
n_akademik : float
n_wawancara : float
n_prestasi : float
n_catatan : text
status_bs : varchar(30)

design_beasiswa tbl_user
user_id : int(11)
user_email : varchar(100)
user_no : varchar(20)
user_password : varchar(100)
user_firstname : varchar(50)
user_lastname : varchar(50)
user_role : varchar(15)
user_status : varchar(15)
user_regtoken : varchar(255)
user_fortoken : varchar(255)
user_create : datetime
user_jur : varchar(100)

design_beasiswa tbl_firebase
id_tkn : int(11)
tkn_firebase : text
tkn_user : varchar(20)
create_at : timestamp

Figure 3 The database application

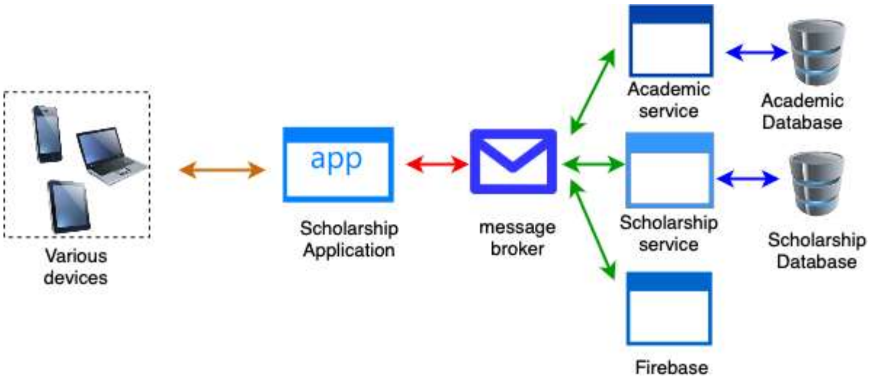


Figure 4 System Architecture

The microservices-based scholarship application architecture involves several important components, including the user device, scholarship application, message broker, and several services and databases. Here is a brief narration of each component and how they interact. The scholarship application can be accessed by users from various devices, such as smartphones, tablets, and laptops. In this architecture, the scholarship application is the core application that serves as the interface between the user and the system. It also sends and receives messages from the message broker. The application (client application) was developed using Veu.js.

Message Broker handles communication between the scholarship application and various services. Messages sent from the application are forwarded to the relevant services, such as Academic Service and Scholarship Service. The message broker helps distribute messages efficiently to ensure that each service receives and processes the appropriate data.

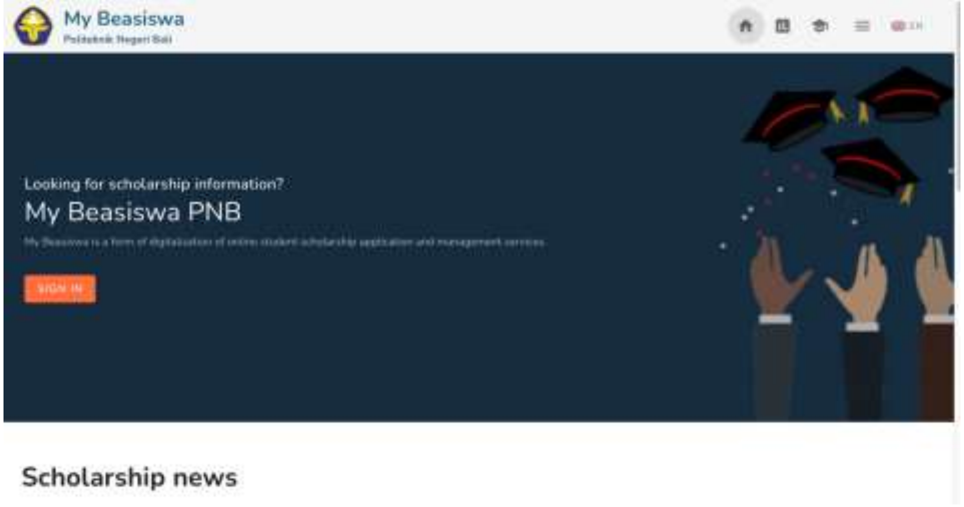
An application built on a microservices architecture operates as a distributed system where multiple processes or services are executed independently. These services are required to communicate with one another through inter-process communication mechanisms. There are two inter-service communication patterns in microservices: synchronous and asynchronous (Microsoft Documentation Website, 2022). In this research, we used an asynchronous inter-service communication pattern, which means the caller does not have to wait for a response from another service, allowing for the elimination of dependencies between services. This non-blocking communication model enhances the overall performance and scalability of the system.

The academic service communicates with the academic database to handle necessary academic information, such as the user's education records that may be relevant for scholarship applications, as well as the scholarship service. This service communicates directly with the scholarship database to manage scholarship-related data, such as criteria, submitted applications, and user application status.

Firebase is used as an additional service, possibly for various purposes such as user authentication, file storage, or push notifications. Firebase is often integrated into cloud-based applications to enhance functionality, such as real-time authentication and data storage. By using this microservices-based architecture, each separate component or service can be managed and developed independently, increasing the scalability, flexibility, and efficiency of system management. Message brokers play an important role in connecting these components, allowing the system to operate in real-time and be well integrated with each other.

4.2 Implementation and system integration results

Figure 5 shows the home page of the scholarship application, and the submission menu is shown in Figure 6, while Figure 7 shows the tracking menu.



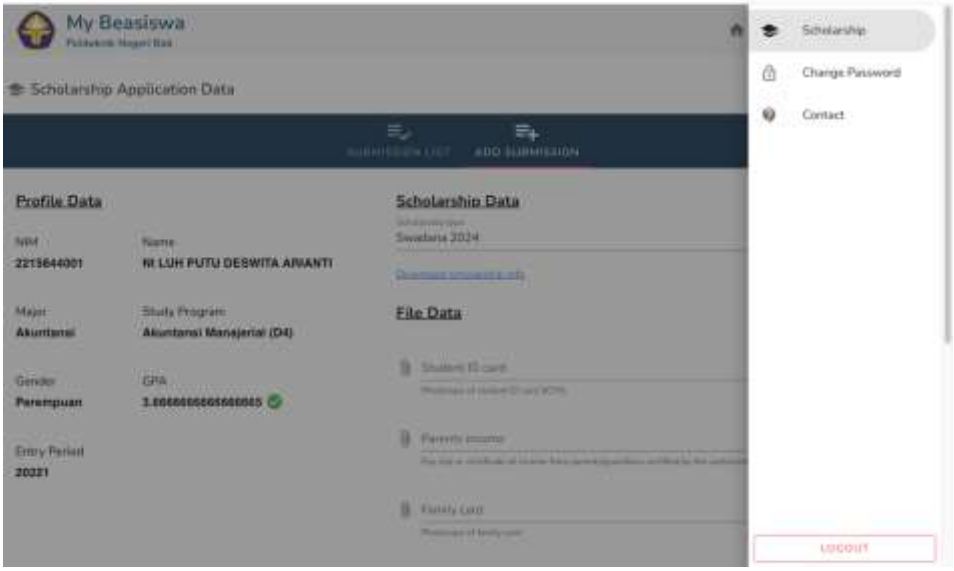
Looking for scholarship information?

My Beasiswa PNB

My Beasiswa is a form of digitalization of online student scholarship application and management services

Scholarship news

Figure 5 The home page of the scholarship application



Scholarship Application Data

Profile Data

NIM
2215644001

Name
NI LUH PUTU DESWITA ARIANTI

Major
Akuntansi

Study Program
Akuntansi Manajerial (D4)

Gender
Perempuan

GPA
3.8666666666666665 ✓

Entry Period
20221

Scholarship Data

Swadana 2024

File Data

Student ID card

Parents income

Family card

Figure 6 The submission menu

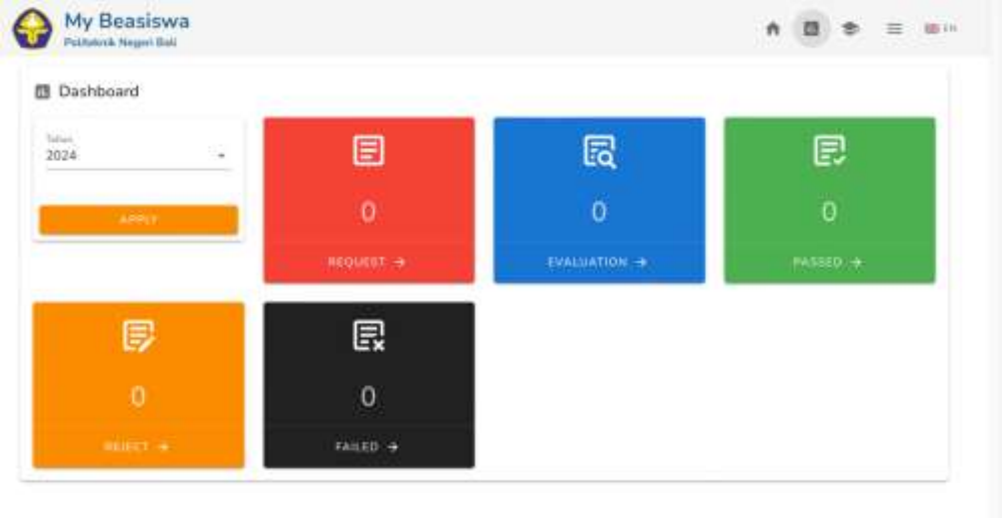


Figure 7 The tracking menu

4.3 User acceptance test results and analysis.

There are 40 students who participated in the User Acceptance Testing (UAT), with a role as the applicants. Students were instructed to use their devices to access the scholarship application as part of the test scenario. They were then instructed to complete a survey on their experiences with the app. Table 3 shows the results of the questionnaire. User Acceptance Testing (UAT) was run during this research, with a Likert scale from 1 to 4 being used for the closed question UAT, where a score of 4 would be considered optimal.

Table 4 Questions list of the questionnaire

Questions	score			
	4	3	2	1
Do you think you need an application to apply for a scholarship?	21	18	0	0
How often would you access the scholarship app if it contained information about scholarships?	14	19	5	2
How easy is it to find the information you need in this app?	14	21	5	0
Does this app fulfil your needs in finding information about scholarships?	14	24	1	0
How satisfied are you with the speed of the app?	9	31	0	0
Do you use this app offline?	1	9	22	8
What do you think about the look and design of this app?	5	31	4	0
Do you find this app helpful in managing your scholarship applications?	21	18	1	0
How often do you receive push notifications from this app?	0	20	20	0
How do you feel overall when using this app?	9	31	0	0

The summary of the conclusions from the questionnaire results table about the scholarship application are as follows:

- Application Scholarship Registration: Most of the respondents (21 respondents scored 4, 18 respondents scored 3) agreed that a scholarship registration application is very important.
- How often do you access the application? The app is accessed very frequently (14 respondents rated 4, 19 respondents rated 3), while there are only a few people who access it infrequently.
- Accessibility of information: Some have problems finding data easily (7 respondents rated 3, 8 respondents rated 2), and many others think that they can find information very easily.
- The Adequacy of scholarship information: Most respondents (14 respondents rated 4, 24 respondents rated 3) felt the app fulfilled their scholarship information needs.
- The application's speed: Satisfaction with the speed of the app was high (9 respondents rated 4, 31 respondents rated 3).
- Offline Usage: Most respondents did not use the app offline (22 respondents rated 2, 8 respondents rated 1).

- App design: Many respondents were satisfied with the appearance and design of the app (5 respondents rated 4, 31 respondents rated 3).
- Usefulness in Scholarship Management: The app is very helpful in managing scholarship applications (21 respondents scored 4, 18 respondents scored 3).
- Push notifications: Respondents were divided on the push notifications experience (20 respondents rated 3, 20 respondents rated 2).
- Overall experience: The majority of respondents felt positive overall (9 respondents rated 4, 31 respondents rated 3).

5. Conclusion

Based on the results obtained from UAT, it can be concluded that the scholarship app was rated positively in terms of usability, access to information, design, and speed. However, improvements are needed to offline usage features and push notification consistency. This indicates that the implementation of PWA in the scholarship application using microservice architecture has worked well. The application performance has been able to overcome problems in terms of scalability, speed, and reliability.

Funding: This research is funded by DIPA of Politeknik Negeri Bali
Conflicts of Interest: The authors declare no conflict of interest.
ORCID iD: 0000-0003-3752-6084

References

[1] Adetunji, O., Ajaegbu, C., Otuneme, N., & Omotosho, O. J. (2020). Dawning of Progressive Web Applications (PWA): Edging Out the Pitfalls of Traditional Mobile Development. Technology and Sciences (ASRJETS) American Scientific Research *Journal for Engineering, 68*(1), 85–99. http://asrjetsjournal.org/

[2] Adetunji, O., Awoniyi, A., & Abel, S. (2023). Evolving Book Sales: From Scrolls to Progressive Web Apps in Online Bookstores. *European Modern Studies Journal,* 7(4), 380–389. https://doi.org/10.59573/emsj.7(4).2023.34

[3] Asri, S. A., Gede, N., Astawa, A., Agung, G., Sunaya, M., Riyan, M., Nugroho, A., & Setiawan, W. (2022). Implementation of Asynchronous Microservices Architecture on Smart Villag*e Application*. *12*(3).

[4] Blinowski, G., Ojdowska, A., & Przybylek, A. (2022). Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation. *IEEE Access*, *10*, 20357–20374. https://doi.org/10.1109/ACCESS.2022.3152803

[5] Bravetti, M., Giallorenzo, S., Mauro, J., Talevi, I., & Zavattaro, G. (2019). Optimal and automated deployment for microservices. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11424 LNCS*, 351–368. https://doi.org/10.1007/978-3-030-16722-6_21

[6] Dwi P P, Adhika P W & Dea S. (2023). Collaboration of Progressive Web App (PWA) And Firebase Cloud Messaging (FCM) for Optimal Performance Mailing Software. *INTERNATIONAL CONFERENCE ON DIGITAL ADVANCE TOURISM, MANAGEMENT AND TECHNOLOGY*, *1*(2), 586–596. https://doi.org/10.56910/ictmt.v1i2.126

[7] Irudayaraj, J. P. (2019). Adoption Advantages Of Micro-Service Architecture In Software Industries. *Article in International Journal of Scientific & Technology Research*. www.ijstr.org

[8] Microsoft Documentation Website. (2022). *Communication in a microservice architecture*.

[9] Nugraha, D., Anjara, F., & Faizah, S. (2022). Comparison of Web Based and PWA in Online Learning. *Proceedings of the 5th FIRST T1 T2 2021 International Conference (FIRST-T1-T2 2021)*, *9*. https://doi.org/10.2991/ahe.k.220205.035

[10] Pikuliak, M., Lazarovych, I., & Usyk, M. (n.d.). PROGRESSIVE WEB TECHNOLOGY-BASED IMPROVEMENT OF THE DISTANCE LEARNING ADAPTIVE SYSTEM. *Scientific Journal of the Ternopil National Technical University*, *2022*(1). https://doi.org/10.33108/visnyk_tntu

[11] Smyk, T., Superson, W., & Plechawska-Wójcik, M. (2023). *Comparative analysis of methods for testing web applications Analiza porównawcza sposobów testowania aplikacji internetowych*.

[12] Swati, J., & Nitin, P. (n.d.). CryptoScholarChain: Revolutionizing Scholarship Management Framework with Blockchain Technology. In *IJACSA) International Journal of Advanced Computer Science and Applications* (Vol. 14, Issue 8). www.ijacsa.thesai.org

[13] Waseem, M., Liang, P., Shahin, M., Di Salle, A., & Márquez, G. (2021). *Design, Monitoring, and Testing of Microservices Systems: The Practitioners' Perspective*. https://doi.org/10.1016/j.jss.2021.111061

[14] Yeh, C. T., Chen, T. M., & Chen, M. C. (2019). A progressive web app for evaluating occupation indicator. *Journal of Internet Technology*, *20*(7), 2217–2223. https://doi.org/10.3966/160792642019122007018