COM 2123

ASSIGNMENT TITLE: SOFTWARE IMPROVEMENT PROJECT

University of Venda

Creating Future Leaders

SURNAME AND INITIALS	STUDENT NUMBER
RAMURUNZI M	23013290
MARUBINI VC	23028867
MAFADZA M	22008339
TSHIOLOLI T	23023248
MANENZHE N	23001393
MORUDU K	23009165
MANETA M	23020881
MUDAU M	22021183
KHAKHU L	23005723
KHAKHU NL	23002218

TASK 1: SOFTWARE SELECT

SOFTWARE SELECTED: TENETECH NSFAS WEBSITE (https://tenetech.co.za/login-portal/)

The Tenetech NSFAS website serves as a critical bridge between students in South Africa and the National Student Financial Aid Scheme (NSFAS). This online platform plays a vital role in facilitating the distribution of bursary funds to countless beneficiaries, ensuring access to education for those who might not otherwise have the opportunity. However, despite its significance, the Tenetech NSFAS website currently suffers from several Limitations that hinder user experience and its overall functionality.

These limitations significantly hinder the user experience for NSFAS beneficiaries. Improving the Tenetech NSFAS website would benefit students by providing a more efficient and user-friendly platform and streamline administrative processes for NSFAS. These limitations include:

Performance Bottlenecks: Imagine a student on the cusp of a new semester, eager to access their NSFAS funds for essential textbooks and accommodation. They log onto the Tenetech NSFAS website, only to be met with sluggish loading times and delayed responses. This scenario, unfortunately, represents a common pain point for many users. The website reportedly struggles to handle high volumes of user traffic, resulting in frustrating lags and delays that can significantly disrupt the financial aid process.

Login Struggles: The login process, often the first hurdle for students seeking to access their accounts, is allegedly fraught with challenges. The inability of the website to accommodate multiple login attempts simultaneously can lead to further delays and frustration. This could be particularly problematic during peak application or disbursement periods, creating a bottleneck that impedes timely access to financial resources.

Time-Consuming Verification: Following a successful login, students encounter another potential roadblock: the verification process. The system reportedly relies on a One-Time Password (OTP) for added security. While security measures are commendable, an OTP delivery time exceeding 5 minutes can be a significant deterrent. This extended waiting period not only disrupts the flow of the user experience but may also discourage students from completing essential tasks on the platform.

Limited Document Upload Options: In today's digital world, flexibility and adaptability are paramount. Unfortunately, the Tenetech NSFAS website allegedly restricts students to uploading documents in a single format. This inflexibility can pose a challenge for students who may possess documents in different formats, forcing them to convert files or potentially hindering their ability to submit required documentation.

Restricted Profile Management: Imagine a student needing to update their contact information due to a change in residence. According to reports, the current system compels students to rely on administrators to modify their profiles. This lack of self-service functionality creates an unnecessary dependency and can lead to delays in updating crucial information. Empowering students to manage their own profiles would streamline the process and enhance user autonomy.

Inconsistent Design philosophy: A website's design should not only be visually appealing but also intuitive and user-friendly. However, the Tenetech NSFAS website is said to suffer from inconsistencies in layout across different pages. This lack of cohesion can be confusing for users navigating the platform. Furthermore, reports of overlapping elements and obscured content suggest a need for a more refined and user-centred design approach.

Impact and Opportunity: The issues outlined above paint a picture of a website in need of improvement. These limitations

not only hinder the user experience for NSFAS beneficiaries but also potentially create additional burdens on administrative staff. By addressing these shortcomings, the Tenetech NSFAS website has the potential to be transformed into a more efficient, user-friendly, and reliable platform. This, in turn, would benefit countless students across South Africa by ensuring smoother access to the financial resources they need to pursue their educational goals.

A Platform with Potential: The Tenetech NSFAS website serves a critical purpose in the South African educational landscape. By improving its functionality and user experience, we can empower students and streamline the financial aid process. Choosing this software application for analysis and improvement presents a valuable opportunity to demonstrate your problem-solving skills, program analysis capabilities, and talent for crafting compelling presentations. By crafting solutions to address the limitations outlined above, you can contribute to a more efficient and user-friendly platform that empowers students and strengthens the foundation of the NSFAS program.

TASK 2: REQUIREMENT ANALYSIS

The Tenetech NSFAS website plays a pivotal role in facilitating financial aid for students in South Africa. This comprehensive requirement analysis will delve into the website's existing functionalities, identify shortcomings that hinder user experience, and propose areas for improvement.

Existing Functionalities:

1. **Account Management:** The website allows students to log in and access their NSFAS accounts. This includes functionalities like viewing bursary balances and potentially tracking disbursement history.
2. **Document Submission:** Students can upload required documents for verification purposes. However, limitations exist regarding the supported document formats
3. **Basic Profile Management:** While the extent is unclear, the website likely allows students to view some profile information.
4. **Communication Channel:** The platform may offer a basic communication channel, potentially through a messaging system or FAQs, to enable students to contact NSFAS for assistance.
5. **Login and Security:** The website utilizes a login system and reportedly implements OTP verification for added security.

SHORTCOMINGS AND AREAS FOR IMPROVEMENT:

Performance and scalability:

1. **Lag and Slow Loading Times:** The website struggles to handle high volumes of user traffic, resulting in lags and delays that disrupt the user experience.
2. **Scalability Issues:** The current infrastructure may not be equipped to handle increasing user demands, potentially leading to system crashes or extended downtime during peak periods.

Login and Authentication:

1. **Limited Login Attempts:** The inability to accommodate multiple login attempts simultaneously creates unnecessary hurdles, especially for users encountering login issues.
2. **OTP delivery time:** The extended waiting period for OTP delivery (reportedly exceeding 5 minutes) significantly disrupts the login process and discourages user engagement.

Document management

1. **Restricted Upload Formats:** The website's inflexibility regarding document formats can create problems for students who possess documents in unsupported formats. This can lead to delays or hinder students from submitting required documentation.

Profile Management

1. **Limited Self-Service Options:** Students reportedly lack the ability to independently edit their profiles, creating a dependency on administrators for even basic updates. This can lead to delays and frustration.

Usability and Design

1. **Inconsistent Layout:** Inconsistent design across different pages can be confusing for users navigating the platform. Overlapping elements and obscured content further detract from the user experience.
2. **Lack of user-friendliness:** The website's design may not be intuitive, potentially creating challenges for users unfamiliar with the platform.

AREA OF IMPROVEMENT:

There's significant room for improvement on the Tenetech NSFAS website. The current system struggles with performance issues, leading to lags and delays during peak usage times. This can be addressed through infrastructure upgrades and performance optimization techniques. The login system also requires attention, with limitations on login attempts and lengthy OTP delivery times creating unnecessary hurdles for students. Furthermore, the website restricts students to uploading documents in a single format, which can be inconvenient if they have documents in other commonly used formats. Additionally, students currently lack the ability to edit their profiles independently, creating a dependency on administrators for even basic updates. Finally, the website's design suffers from inconsistency across different pages, potentially confusing users and hindering navigation. By implementing a user-centered design approach that prioritizes clarity and intuitive navigation, the user experience can be significantly improved. These are just some of the key areas that warrant improvement to create a more efficient, user-friendly, and reliable platform for NSFAS beneficiaries. The objectives of the proposed improvements for the Tenetech NSFAS website includes the ones below

Enhanced with user experience:

1. **Reduce wait times and frustration:** This includes improving website performance to eliminate lags and delays, streamlining the login process by allowing multiple login attempts and significantly reducing OTP delivery times.

2. **Increase flexibility and convenience:** Expanding supported document upload formats provides students with more options. Empowering students with self-service profile editing functionalities reduces their reliance on administrators for basic updates.
3. **Improve clarity and navigation:** Implementing a user-centered design approach with consistent layouts across all pages will make the website easier to navigate and understand, minimizing confusion for users.

Increased efficiency and scalability:

1. **Handle higher traffic volumes:** Upgrading the website's infrastructure will ensure it can accommodate a larger number of users without experiencing performance issues.
2. **Reduce administrative burden:** Empowering students with self-service functionalities like profile editing can free up administrator time to focus on other tasks.

Improved accessibility and inclusivity:

1. **Multilingual support:** Catering to a wider audience by offering the website in multiple languages ensures inclusivity for students with diverse backgrounds.
2. **Accessibility features:** Integrating features that cater to individuals with disabilities allow everyone to access the platform and manage their NSFAS applications seamlessly.

TASK 3: SOLUTION PROPOSAL

Building upon the identified shortcomings and areas for improvement, this section will delve into proposing concrete and implementable solutions for the Tenetech NSFAS website.

1. Performance and Scalability:

Solution: Migrate the Tenetech NSFAS website to a cloud-based infrastructure provider like Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform (GCP). These platforms offer robust, scalable solutions with features like auto-scaling.

Implementation: Partner with a cloud service provider to conduct a thorough assessment of the website's current resource requirements (CPU, memory, storage). This assessment will form the foundation for designing a scalable cloud architecture. The chosen cloud platform offers pre-configured components like auto-scaling groups that can be leveraged. These groups dynamically adjust server resources based on real-time user traffic data retrieved from website monitoring tools. For instance, AWS offers features like Amazon CloudWatch that provide real-time insights into user traffic patterns. By integrating the website with CloudWatch, auto-scaling groups can be configured to automatically scale up server resources during peak usage times and scale down during periods of low activity. This approach ensures optimal performance and resource utilization, eliminating website crashes due to overloaded servers.

2. Login and Authentication:

Solution 1: Integrate Multi-Factor Authentication (MFA) options alongside usernames and passwords. This can be achieved through partnerships with mobile authentication providers like Google Authenticator or Microsoft Authenticator.

Implementation: The website can leverage industry-standard libraries or APIs offered by these mobile authentication providers to integrate MFA functionality. This would involve integrating the provider's SDK (Software Development Kit) into the website's codebase. The website would then guide users through a one-time setup process where they link their mobile devices with the NSFAS platform for MFA. During login, after entering their ID, phone number and password, users would be prompted to approve the login attempt from their mobile device using a one-time code received via SMS. This two-factor approach significantly strengthens login security while still allowing for multiple concurrent login attempts from the same device, eliminating unnecessary hurdles for legitimate users.

Solution 2: Implement a queueing system for OTP delivery during peak usage times. This system should prioritize critical messages like OTPs based on urgency and user roles.

Implementation: The current system responsible for OTP delivery can be integrated with a message queuing service offered by the chosen cloud platform (e.g., Amazon Simple Queue Service (SQS) in AWS). This service acts as a buffer, managing a queue of outgoing messages. Messages are processed on a first-in, first-out basis by worker processes. However, the queuing system can be configured to prioritize critical messages like OTPs based on pre-defined rules. This prioritization ensures that even during peak periods when the message queue is full, OTPs are delivered promptly, minimizing wait times for students attempting to log in.

3. Document Management:

Solution 1: Integrate a document conversion tool within the website. This can be achieved using open-source libraries like Apache POI or LibreOffice.
Implementation: The website's backend code can be integrated with libraries like Apache POI to enable real-time document conversion. Uploaded documents would be intercepted by the server-side code before being stored in the system. Apache POI provides functionalities to read and convert various document formats (e.g., DOCX, XLSX) into a standardized format (e.g., PDF) that is compatible with the website's internal processes. This conversion occurs seamlessly in the background, without any additional steps required by the user.
Solution 2: Implement a real-time document preview feature using JavaScript libraries like FileReader API.
Implementation: The website's user interface can leverage the FileReader API to allow users to select and preview their documents before uploading. This functionality would provide immediate feedback on file format compatibility. The user would be able to see a representation of their document within the browser window. The FileReader API would also provide information about the file size and format. This allows users to identify any issues with their documents (e.g., incorrect format, exceeding size limits) before they are uploaded to the system. This approach minimizes errors and rework, streamlining the document submission process for students.

4. Profile Management:
Solution: Develop a secure self-service profile management portal. This portal should leverage role-based access control (RBAC) and utilize secure communication protocols like HTTPS with data encryption at rest and in transit.

Implementation: A dedicated section within the website can be developed as a secure self-service portal. Role-based access control (RBAC) would be implemented to restrict functionalities based on user roles (e.g., student, administrator). Students would have access to functionalities like updating their contact information.

5. Usability and Design:

Solution 1: Conduct user testing sessions with NSFAS beneficiaries to gather feedback and inform a user-centered redesign process.

Implementation: Partner with user experience (UX) specialists to design and conduct user testing sessions. A representative group of NSFAS beneficiaries would be recruited to participate in these sessions. During the sessions, users would be tasked with navigating the website and completing specific actions (e.g., applying for funding, and uploading documents). Their interactions would be observed and recorded, and they would be encouraged to provide feedback on their experience. This feedback would then be analyzed by the UX team to identify areas for improvement in the website's usability, layout, and information architecture.

Additional Considerations: The user testing sessions can be conducted in various formats, including in-person sessions, remote online sessions, or a combination of both. Think-aloud protocols can be employed where users verbalize their thoughts and thought processes while navigating the website. This can provide valuable insights into user behavior and thought patterns.

Solution 2: Implement responsive web design (RWD) principles to ensure the website adapts seamlessly to different screen sizes and devices.

Implementation: The website's codebase can be refactored using responsive web design (RWD) principles. This ensures that the website layout adjusts dynamically based on the screen size of the device being used (desktop, laptop, tablet, smartphone). Responsive design frameworks like Bootstrap or Foundation can be utilized to streamline this process. These frameworks provide pre-built components and styles that automatically adjust to different screen sizes. By implementing RWD, the website will be accessible and user-friendly regardless of the device a student is using, ensuring a consistent experience across all platforms.

TASK 4: ALGORITHM DESIGN

1. Performance and Scalability

Pseudocode

Migrate To Cloud And Scale Website (website_requirements)

PARTNER with a cloud service provider (e.g., AWS, Azure, GCP)

ASSESS the current resource requirements of the website

GET the website's CPU, memory, and storage requirements

DESIGN a scalable cloud architecture based on the assessment

CHOOSE the appropriate cloud platform and services

CONFIGURE the cloud infrastructure:

CREATE auto-scaling groups

INTEGRATE website monitoring tools (e.g., CloudWatch)

CONFIGURE auto-scaling rules based on real-time user traffic data

SET triggers for scaling up and scaling down resources

MIGRATE the website to the cloud-based infrastructure

MONITOR the website's performance and user traffic patterns

RETRIEVE real-time user traffic data from the monitoring tools

WHILE TRUE:

IF user traffic increases:

AUTO-SCALE the resources by adding more servers to the auto-scaling groups

ELSE IF user traffic decreases:

AUTO-SCALE the resources by removing servers from the auto-scaling group

OPTIMIZE the cloud infrastructure and auto-scaling rules based on ongoing performance monitoring

ADJUST auto-scaling rules and thresholds as needed

DELAY for a specified monitoring interval

END Session

2.Login and authentication

Pseudocode

LoginProcess()

DISPLAY login page

GET user's ID and phone number

IF user's login details are valid

PRINT "Verified"

GET user's password

IF password is correct:

GENERATE one-time password (OTP)

SEND OTP to user's phone number

GET user's OTP input

IF OTP is correct:

GET user's 4-digit passcode

IF passcode is correct:

DISPLAY dashboard (homepage)

ENABLE access to account management features:

- My Account

- Pay A QR Code

- GET Paid With Your QR Code

- Make An EFT

- Buy Airtime And Data

- Transaction History

ALLOW user to choose desired action

ELSE:

PRINT “Please enter correct code”

ELSE:

PRINT “Invalid OTP”

ELSE:

PRINT “Invalid credentials”

ELSE:

IF user ID not found:

PRINT “ID not found”

ELSE:

PRINT “Phone number cannot be found”

RETRIEVE user’s profile information

DISPLAY user’s contact details (phone number)

ENABLE editing of contact details

GET new contact details from user

GENERATE one-time code

SEND one-time code to user’s contact

GET one-time code input from user

IF one-time code is correct:

UPDATE user’s contact details in the system

PRINT “Contact information updated successfully”

ELSE:

PRINT “Invalid one-time code”

3.Document management

Pseudocode

DocumentManagement()

After user’s login details are validated

DISPLAY dashboard (homepage)

ENABLE access to account management features(more)

DISPLAY user’s profile

IF KYC (Know Your Customer) process has failed:

SCROLL DOWN to the “Upload Document” section

ALLOW user to select a document to upload

TRIGGER the FileReader API to check the content of the selected document

DISPLAY a preview of the selected document in the browser

RETRIEVE information about the file (size, format)

DISPLAY the document size and format to the user

IF the document format or size is not compatible:

PRINT “Error”

ALLOW user to select a different document or cancel the upload process

ELSE:

ALLOW the user to upload the document

PRINT “Document successfully uploaded”

End Session

4.Profile management

Pseudocode

SelfServiceProfileManagement()

After user has logged in successfully

DISPLAY user's profile page

DISPLAY user's contact information

ENABLE editing of contact information

WHILE user is editing contact information:

GET new contact information from user

GENERATE one-time code

SEND one-time code to user's new contact information

DISPLAY one-time code verification page

GET one-time code from user

IF one-time code is correct:

UPDATE user's contact information

DISPLAY "Contact information updated successfully"

BREAK

ELSE:

DISPLAY error message

End session

5.Usability and Design

Pseudocode

ConductUserTestingSessions(NSFAS_beneficiaries)

PARTNER with user experience (UX) specialists

RECRUIT a representative group of NSFAS beneficiaries to participate

FOR each participant:

OBSERVE and RECORD the participant's interactions while they navigate the website and complete specific tasks

ENCOURAGE the participant to provide feedback on their experience

END FOR

ANALYZE the feedback gathered from tarchitectur

IDENTIFY areas for improvement in the website's usability, layout, and information architecture

INCORPORATE the findings into the redesign process

END Session

Pseudocode IMPLEMENT RESPONSIVE WEB DESIGNS

REFACTOR the website's codebase using responsive web design (RWD) principles

UTILIZE a responsive design framework (Bootstrap, Foundation)

FOR each website element:

ENSURE the layout automatically adjusts based on the screen size of the device

APPLY the appropriate styles and components from the responsive design framework

VERIFY that the website maintains a consistent and user-friendly experience across all devices (desktop, laptop, tablet, smartphone)

End session

TASK 4: VISUALIZATION USING DRAW.IO