

**Review of Mambu Apps**

|  |  |  |
| --- | --- | --- |
| OpinionPencil.png | CalnedarStart.png | CalendarEnd.png |
| **Conclusion/Rating** | **Assignment Fieldwork Start Date** | **Assignment Target Completion** |
| N/A | May 25, 2023 | May 26, 2023 |

Table of Contents

Background and Scope 3

Finding Summary 3

[01 Django debug mode set to true allowing unauthorized access to apps and urls 4](#_TOC_250003)

[Annexures 6](#_TOC_250002)

[Figure 1 6](#_TOC_250002)

[Figure 2 7](#_TOC_250002)

[Figure 3 8](#_TOC_250002)

[Figure 4 9](#_TOC_250002)

[Figure 5 1](#_TOC_250002)0

[Figure 6 1](#_TOC_250002)1

[Figure 7 1](#_TOC_250002)2

[Figure 8 1](#_TOC_250002)3

[Figure 9 1](#_TOC_250002)4

[Figure 10 1](#_TOC_250002)5

Background and Scope

**Background**

Mambu System usually allows for the extension of its default capabilities by giving clients the ability to integrate their own applications directly into Mambu within an embedded iframe. In this context an iframe is used to display external content from another app within a specific section of a web page. Once defined and installed, a Mambu System app is a selectable frame in the Mambu UI that renders an externally-hosted application.

While on another assignment, internal audit noted that the CS settlement app in Platinum Kenya generated an error which on further investigation led to this URL https://webapp.platinumcredit.co.ug/loans/.

**Scope**

The objective of this review is to assess the configuration and handling of Django debug mode and error messages within the organization’s web application environment. The focus is on identifying and addressing vulnerabilities related to the exposure of sensitive information, specifically passwords, in error messages and stack traces.

Finding Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Finding ID** | **Title/Headline** | **Owner** | **Severity** | **Remediation Deadline** |
| 01 | Django Debug Mode Set to True Allowing Unauthorized Access to Apps and URLs | Head of MIS | High |  |

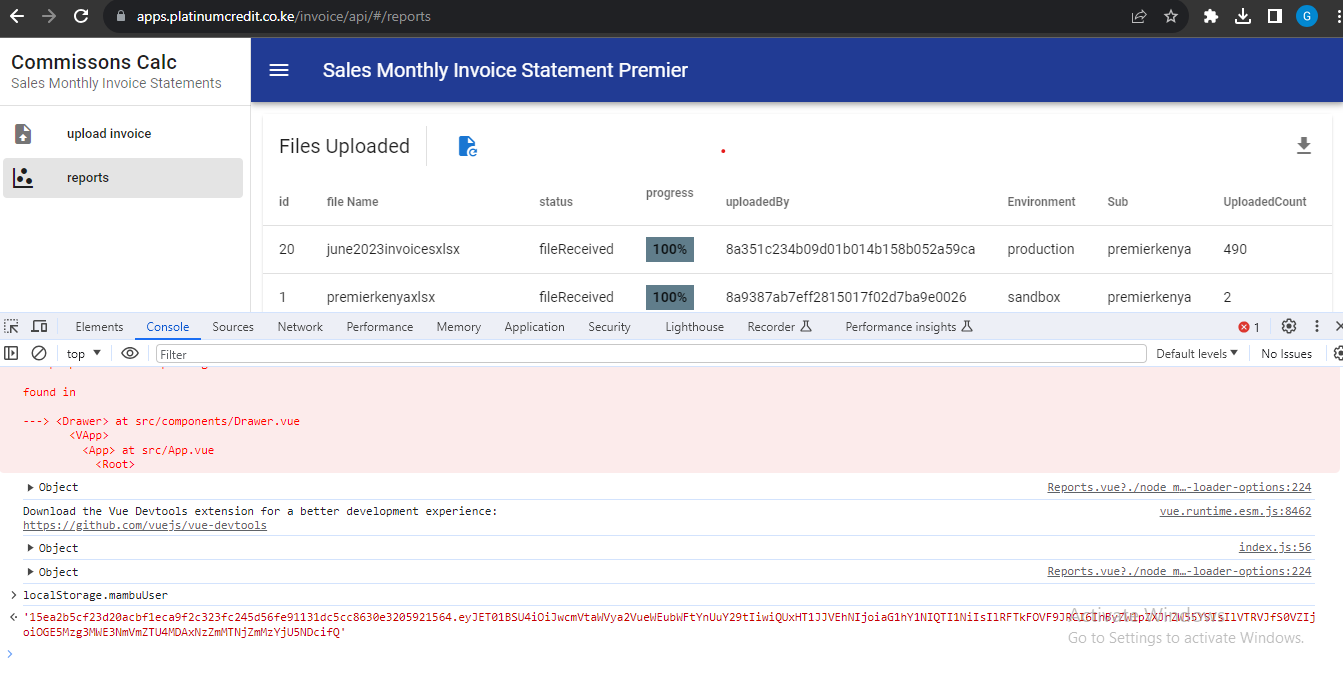
# 01 Django Debug Mode Set to True Allowing Unauthorized Access to Apps and URLs

|  |  |
| --- | --- |
| **Process:** | **Process Owner:** |
| **Owner:** Head of MIS | **Executive Owner:** Managing Director |

|  |  |  |
| --- | --- | --- |
|  | **Finding Information** |  |
| **Overall Status** | **Severity** | **Escalation Level** |
| Open | High |  |
| **Date Identified** | **Remediation Deadline** | **Remediation Status** |
| May 25, 2023 | Immediate |  |
| **Description** |  |  |
| This issue highlights a security vulnerability observed within the Django web framework, where the debug mode has been set to “True”. This configuration allows unauthorized users to access sensitive application information, including URLs, code, and other potentially exploitable details.  During our review, we noted that we could access various URLS as seen in figure 1. We forced the system to generate error messages and stack traces. Through the generated error messages, we noted that we could do the following:   * Get user name and password for the Refinance (Platinum Kenya), agentupdateapi (Platinum Kenya), bulkposting (Platinum Uganda) and journalapi(MCL) users which have API access and are active in Mambu as seen in figure 6 and figure 7 for the Refinance user, and, figure 8 and figure 9 for the bulkposting user, figure 10 for the journalapi user, figure 11 for the agentupdateapi. The refinance calculator in PCLK uses the Refinance user to authenticate in Mambu and is used by credit staff to generate the refinance amount and statement. Also, through this account, unauthorized users can edit loan accounts, edit clients, write off loan accounts, enter repayments, view client details and view loan account details in Mambu (Platinum Kenya). * Generate settlements receipts documents and attach in Mambu (Platinum Kenya). To do this an unauthorized user would need to have the loan id (see figure 2 and figure 3) e.g. [https://webapp.platinumcredit.co.ug/cs/receipt?accountid={lid}&employer=TSCK&firstname=Leo&lastname=cleo](https://webapp.platinumcredit.co.ug/cs/receipt?accountid=1000206739&employer=TSCK&firstname=Leo&lastname=cleo). We further noted that the API user used to generate documents is called clearance and hasn’t been used since 2019. * Generate demand one letter, demand two letter, guarantor one, guarantor two and recall documents and attach in Mambu (Premier Kenya). To do this, an unauthorized user would need to have the loan id and client id of a client in Mambu(see figure 4 and figure 5) e.g. [https://webapp.platinumcredit.co.ug/premier/guarantor\_two?clientid={clientid}&accountid={lid}](https://webapp.platinumcredit.co.ug/premier/guarantor_two?clientid=25657696&accountid=900077743). We further noted that the API user used to generate documents is called clearance and hasn’t been used this year. We further noted that the API user used to generate documents is called DEP API and hasn’t been used since November 1, 2021.  |  |  |  |  | | --- | --- | --- | --- | | **Entity** | **API Name** | **Functionality** | **Date Last used** | | Platinum Kenya | Refinance User | Used by credit staff to generate the refinance amount | Actively in use | | Platinum Kenya | Agentupdateapi | Assigned to the commissions calculator role, which was used to authenticate the initial commissions calculator app. |  | | Platinum Kenya | Clearance | used to generate clearance letters | 12/3/2019 11:09:25 AM | | Premier Kenya | DEP MAILING USER | Used to generate and attach demand and recall letters in Mambu | 11/1/2021 4:07:45 AM |  * **Unauthorized access to the security register through the security register app.** Through these links we could access Premier and Platinum Kenya’s security register: <https://commissions.premiergroup.co.ke/premierkenyasecurityregister/generelReport/>, <https://commissions.premiergroup.co.ke/premierkenyasecurityregister/requestReleaseDocuments/>,  <https://commissions.premiergroup.co.ke/premierkenyasecurityregister/releaseddocu/>, <https://commissions.premiergroup.co.ke/premierkenyasecurityregister/pendingApprovalDocuments/>, <https://commissions.premiergroup.co.ke/premierkenyasecurityregister/rejectedDocuments/>,[https://commissions.premiergroup.co.ke/pclKeSecurityRegisterV2/securities\_general\_report/](https://protect-za.mimecast.com/s/ZXx8Czm5LPUnov2Gig2GTb?domain=commissions.premiergroup.co.ke/) and [https://commissions.premiergroup.co.ke/pclke\_security\_register/securities\_general\_report/](https://protect-za.mimecast.com/s/oqX9CAno0qcVAMWZiYE0A-?domain=commissions.premiergroup.co.ke/) can be accessed through the public internet and exposes personally identifiable information (PII). PII which can be accessed from the above links include client ID numbers, branch, names and staff ID numbers which are part of their display names. Management should ensure that authentication of the application is designed to ensure that it authenticates Mambu session and the role of the user requesting access to the application. We notified the IT Managers on 8th September, 2023, however, PKL’s remain unresolved. The company should ensure applications developed by the group Dev. team implements proper access controls and authentication. Refer to the below image.      * **Unauthorized access to journals through journal upload application** - <https://commissions.premiergroup.co.ke/journal_checker/journal_posted_report_premierke/> (PKL) and <https://commissions.premiergroup.co.ke/platinumKeJournalchecker/journal_posted_report_platke/> (PCLK) These URLs exposes journals which contain sensitive data that include staff mobile numbers, their ID numbers and company data. We notified IT Managers on 8th September and as of 21st Sept, 2023 the issue had been resolved.      * Database administration page of the company website has been left exposed to the public internet which first prompts a user to authenticate themselves through the [url](http://goog_917025368/)<https://premierkenya.co.ke/phpmyadmin>. Management should ensure that only select IP addresses are redirected to the authentication module and the rest get a forbidden or unauthorized access error. We notified the IT Manager on 8th Sept, 2023, however, the issue has not been resolved.      * **Unauthorized access to bulk operation applications embedded to Mambu**   Unauthorized access to applications embedded to Mambu which perform various bulk operations namely; bulk client reassignment, receipt matching, bulk posting, bulk withdrawal mpesa, sales commission calculator. A user can generate a template which guides on fields to be filled with what entries before uploading the template to the application to perform the desired operation. Management should ensure there is a user management module. As an example refer to the image below for client reassignment application. A user has to be logged in to Mambu to be able to use these applications. | | |

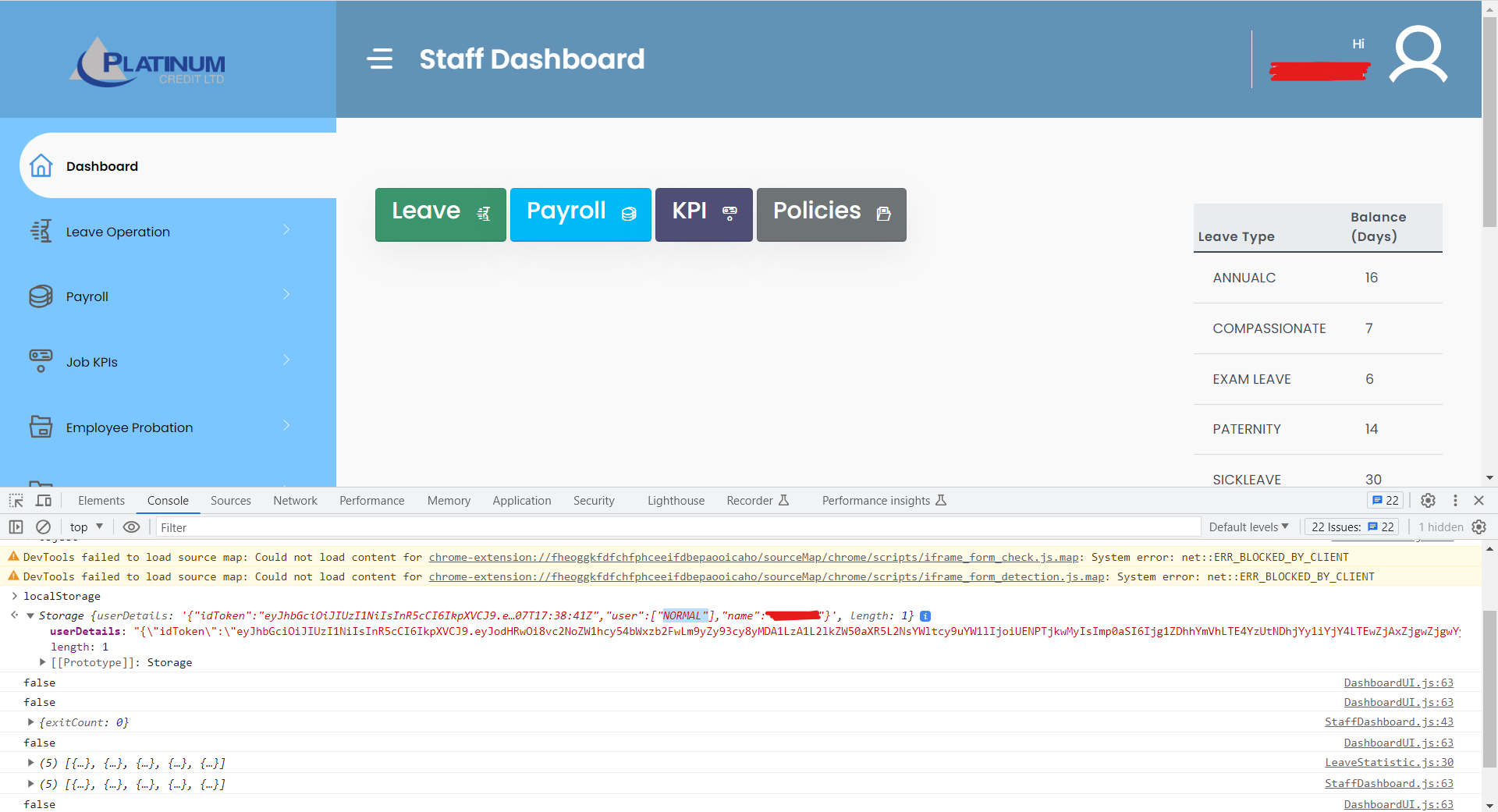
* **Lack of proper access controls on Group applications developed in JavaScript**

We noted a trend where new apps in Mambu are developed using JavaScript frameworks unlike older apps which were built on Django (a python framework).We identified significant security risks related to the storage of authentication tokens and user roles in the client-side localStorage. On reviewing the data stored in the storage we noted the application saves the Mambu encoded key (on creation of a user, Mambu generates encoded key, which is unique to each user). Applications in use by Platinum and Premier Kenya developed using JavaScript and have this weakness include; Sales commission invoice workaround, Receipt matching, PLKE BULK LGF DEPOSIT, SMS Reports, Bulk User Creation, TSC Bulk posting and SME Refinance reports. Development team should ensure authentication tokens are not saved on the client side and the applications maintain proper access controls. Refer to the example below of Sales commission invoice workaround image.

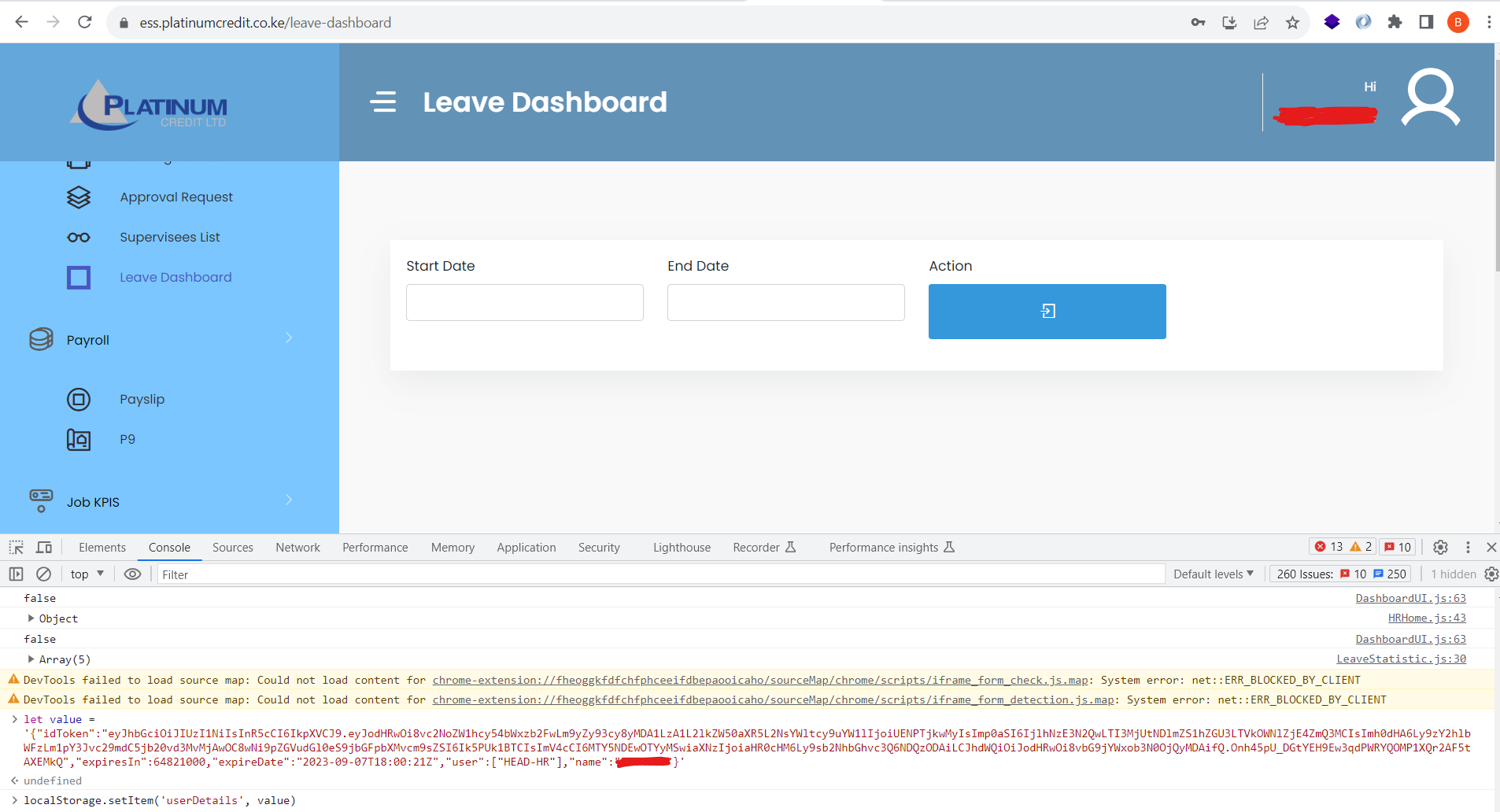


* **Security Risks Associated with Storing Tokens and User Roles in localStorage in ESS**

In assessing the security measures of the ESS app, we identified significant security risks related to the storage of authentication tokens and user roles in the client-side localStorage. On reviewing the data stored in the storage we noted the application saves the username and role in local Storage

 .

On assessing the code under the sources tab, we noted the app had the following roles: NORMAL, HR, HOD-HR, HOD-ICT, HOD-ADMIN, HOD, HEAD-HR, HOD-FIN, and MD. We changed the role to "HOD-HR" from "NORMAL" and the system allowed access to more data such as a report for users in the system, the job seekers list, the leave dashboard, the contracts approval, and the bucketed contracts list as well as the ability to assign documents to staff.

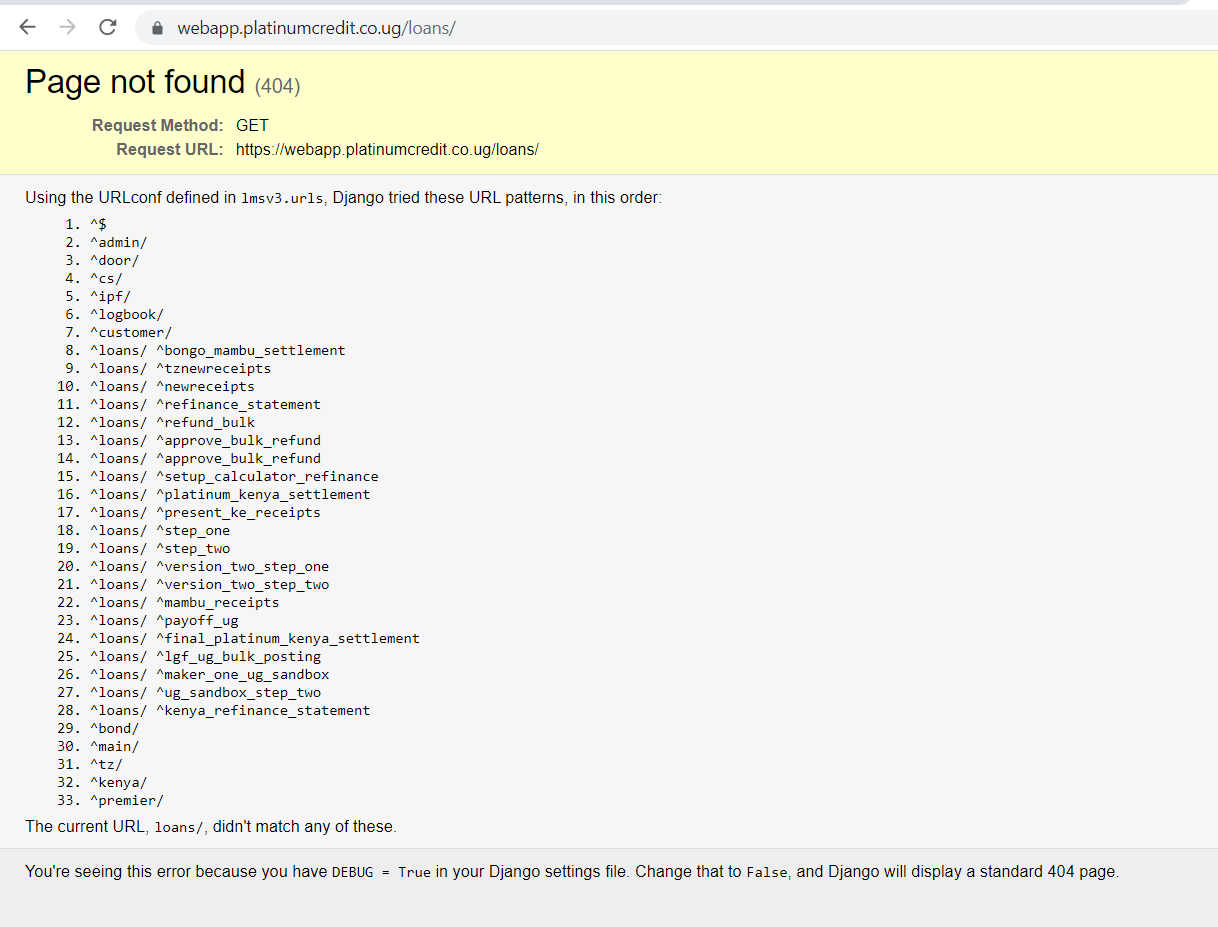


Our recommendation is for the system not to use client-side storage to store sensitive information.

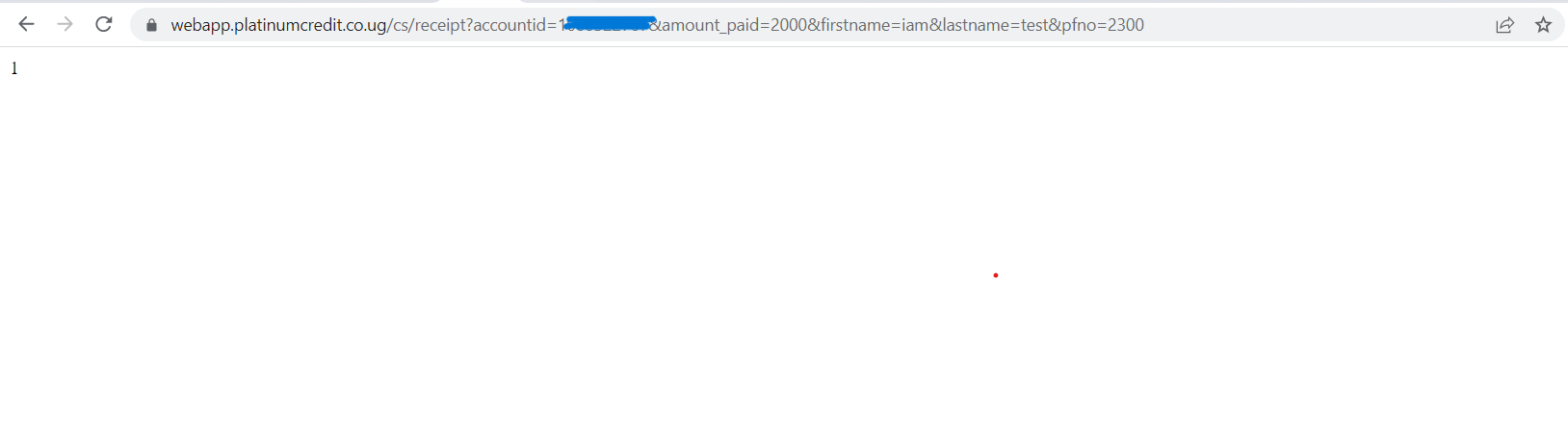
|  |  |
| --- | --- |
| **Risk/Impact** |  |
| Unauthorized access to Mambu. | |
| **Recommendation** | **Remediation Plan/Management Response** |
| Ensure error messages don’t include information such as passwords through secure error logging processes.  Apps that are not in use in Mambu should be deactivated as well as the accounts they use to authenticate.  The passwords for the refinance, journalapi(MCL) and bulkposting users should be changed immediately.  The DEPAPI(Premier Kenya) and clearance(Platinum Kenya) user should be deactivated if they are not in use.  Management should ensure that there is an isolated sandbox environment which will be used to test the applications.  Management should ensure that credentials are not stored on the client-side. |  |

Annexures

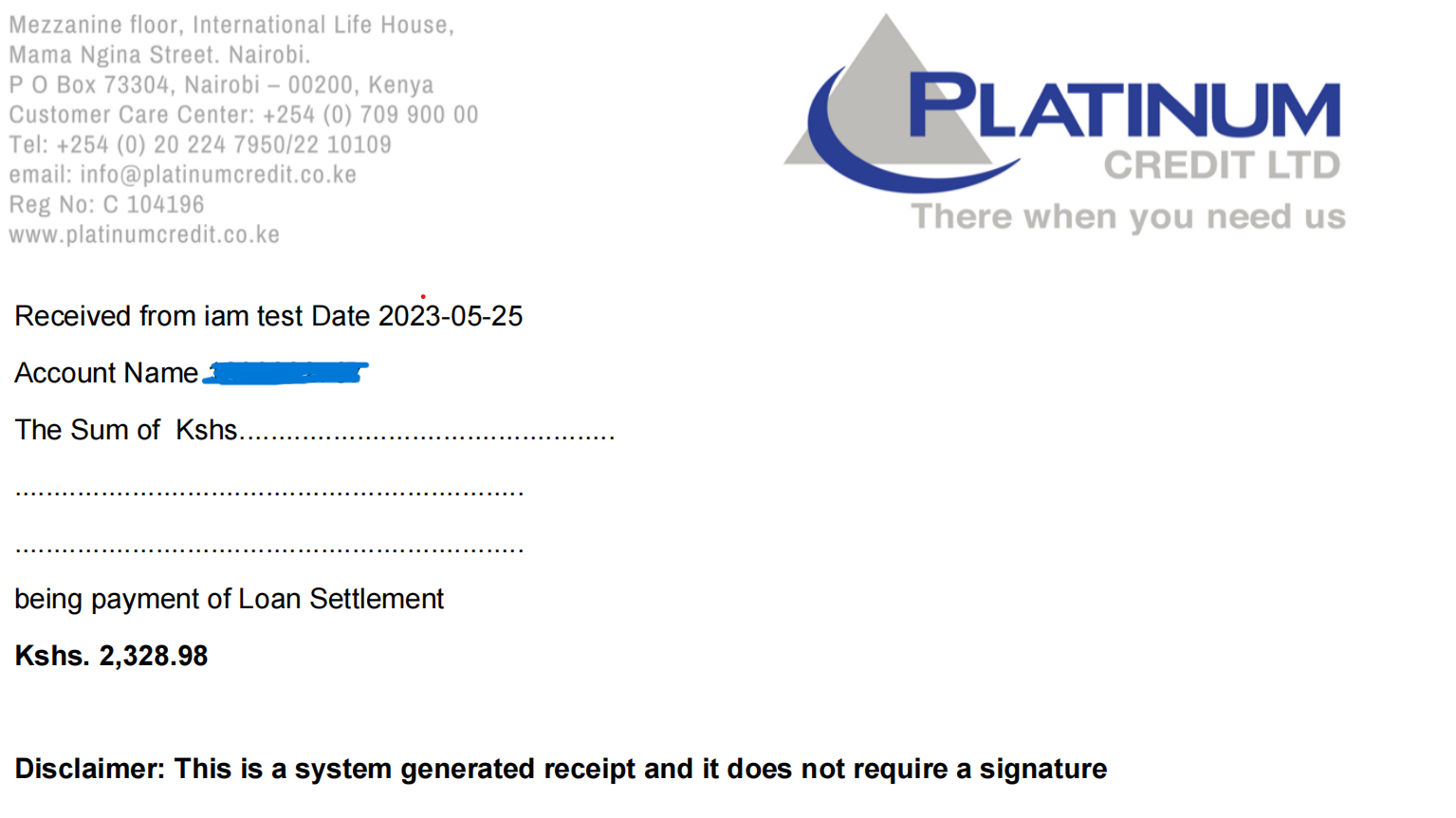
***Figure 1 - Apps and URLs***

**

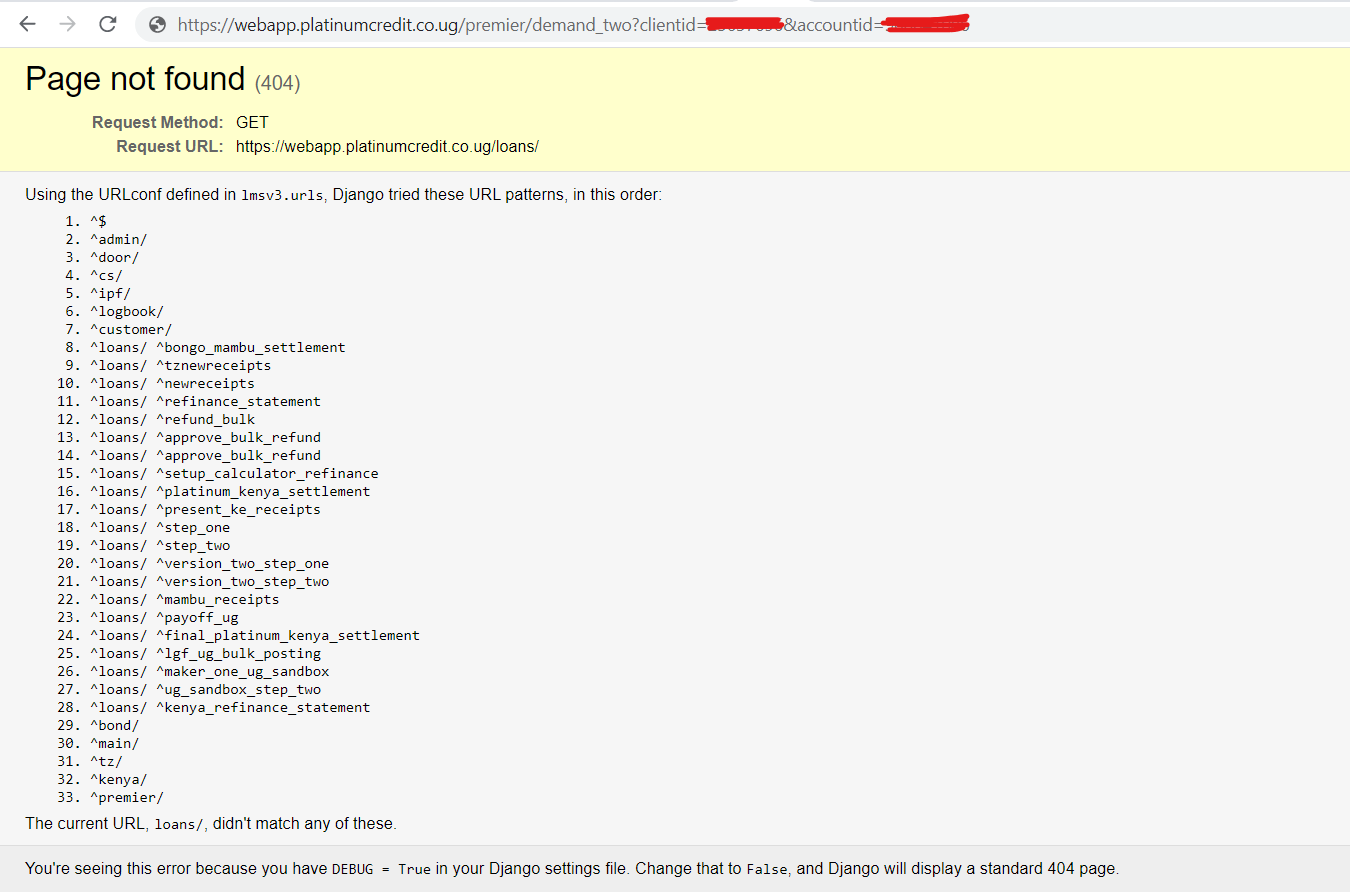
***Figure 2 - Generation of the settlement receipt document, PCLK***

**

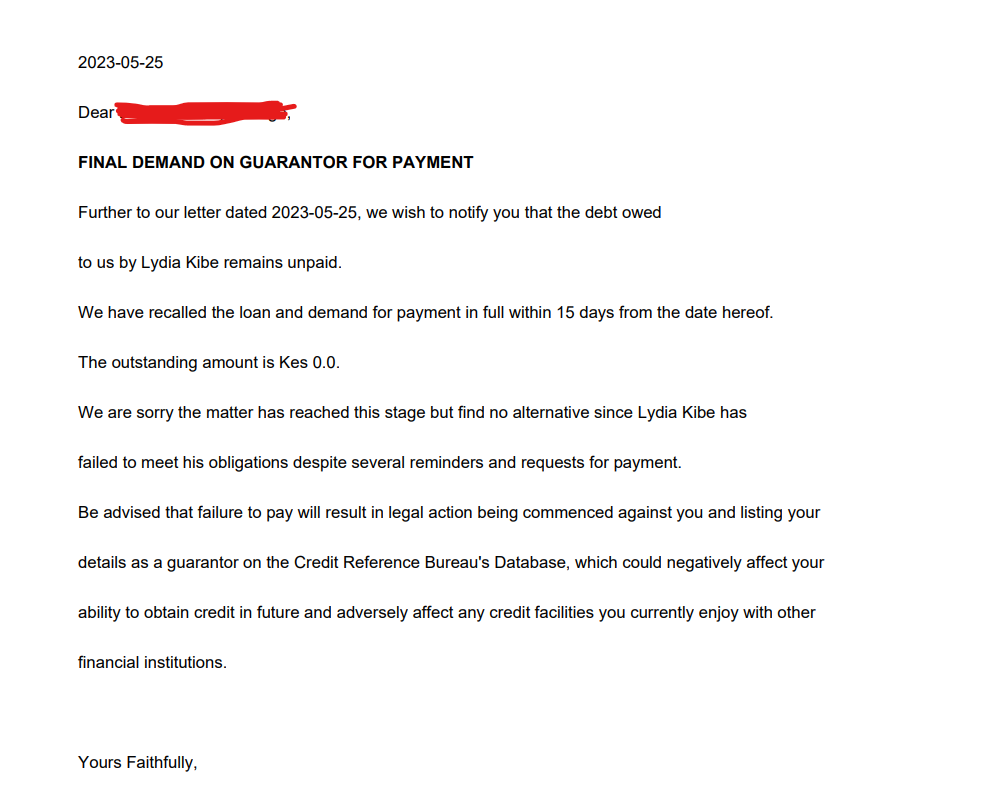
***Figure 3 - PCLK Settlements Receipt***

**

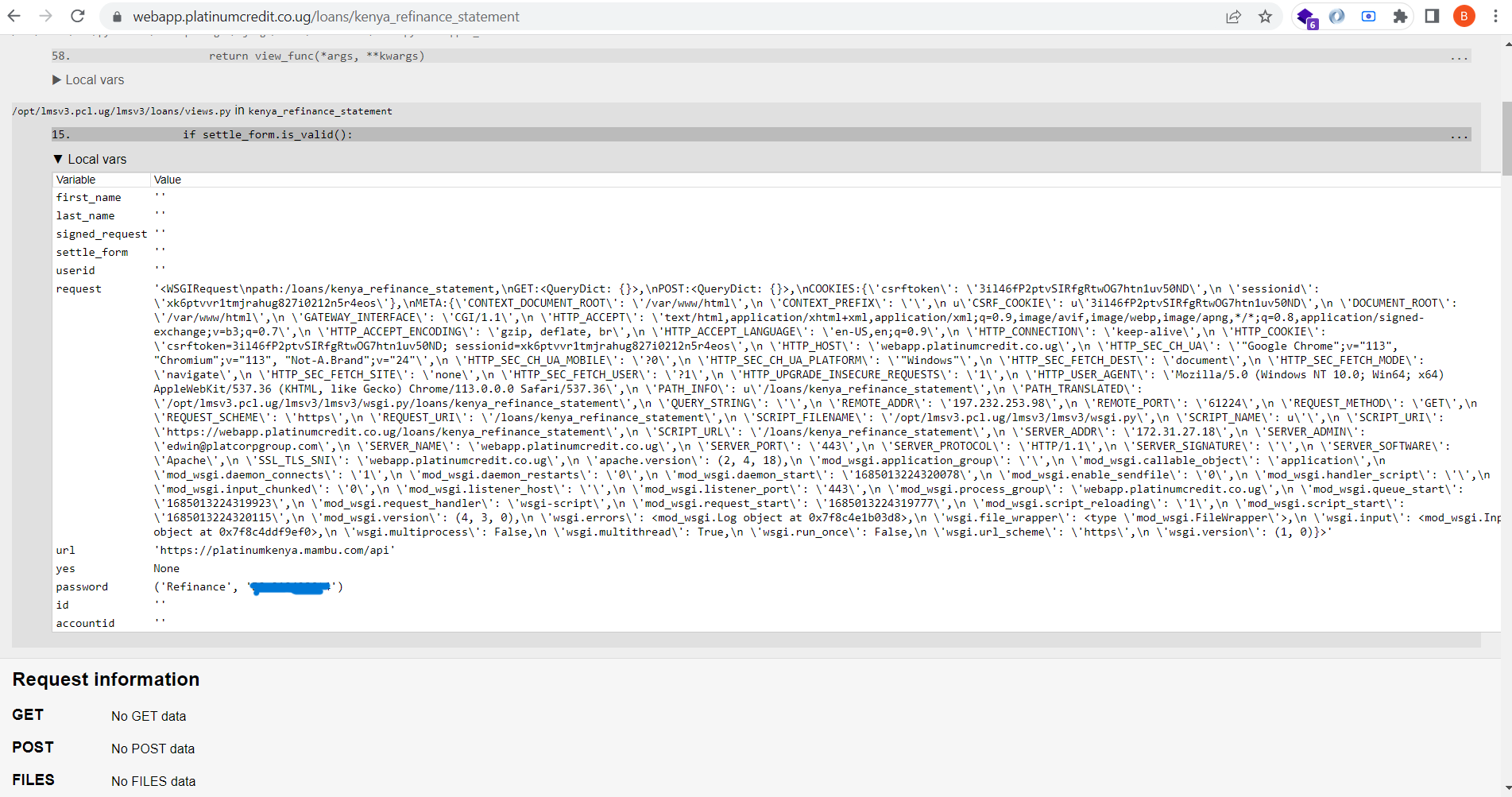
***Figure 4 - Premier Kenya Generate Demand Letter Document***

**

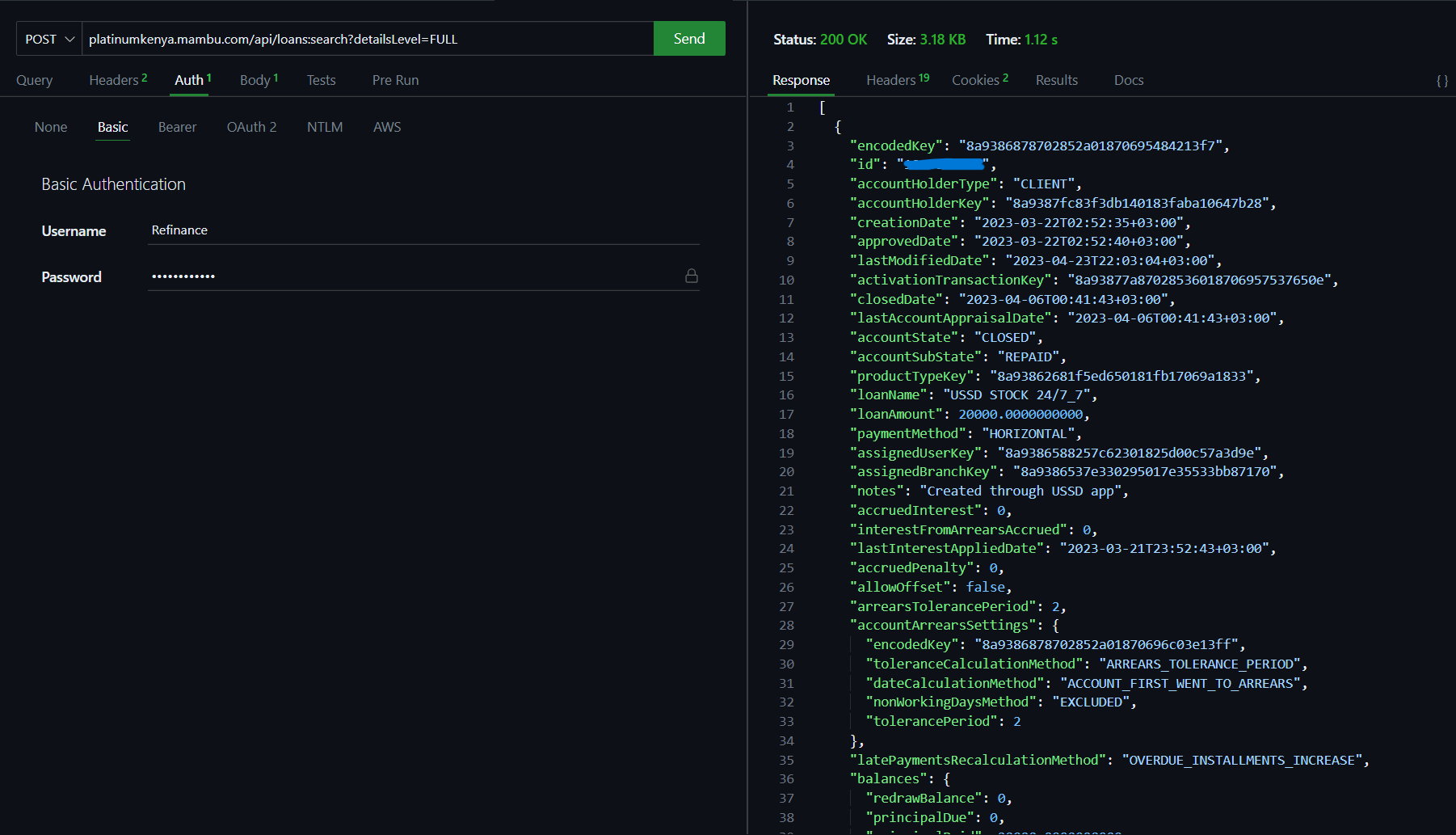
***Figure 5 - Generated Premier Kenya Demand Letter***

**

***Figure 6 - Error message revealing the Refinance user account and password(Platinum Kenya)***

**

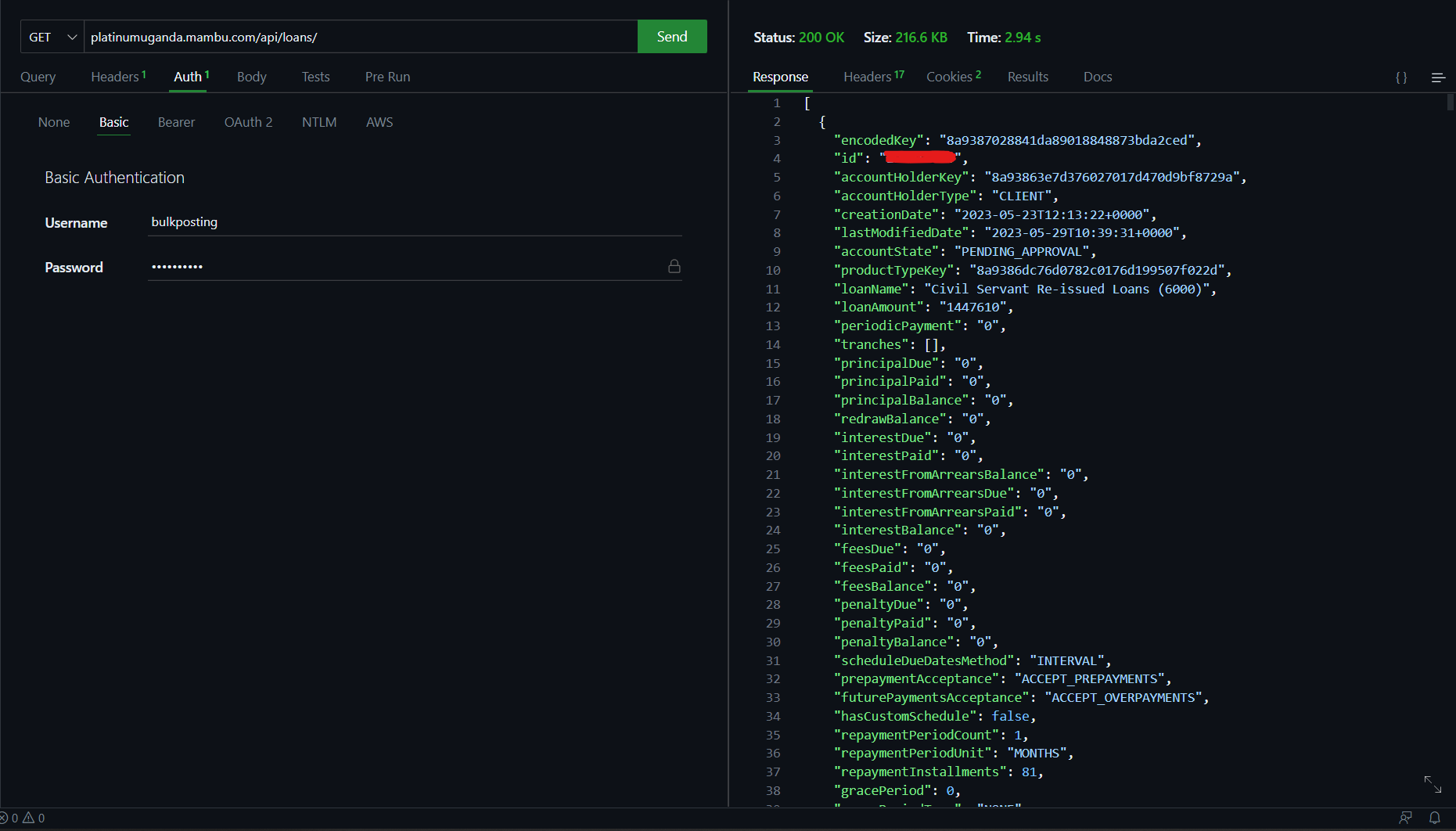
***Figure 7 - Querying Mambu using the Refinance User(Platinum Kenya)***

**

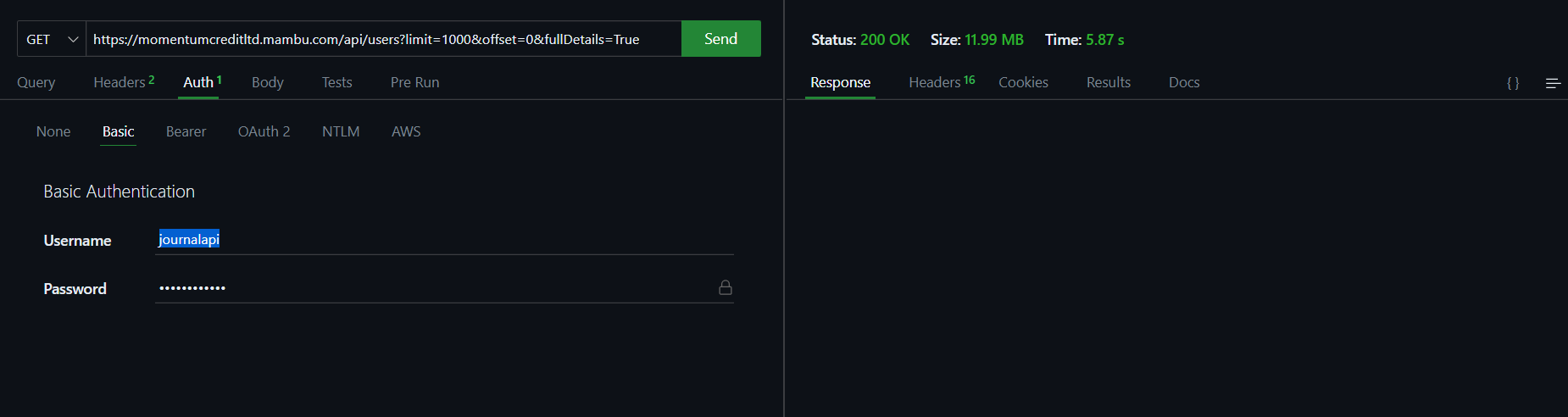
***Figure 8 - Error message revealing the bulkposting user account and password(Platinum Uganda)***

**

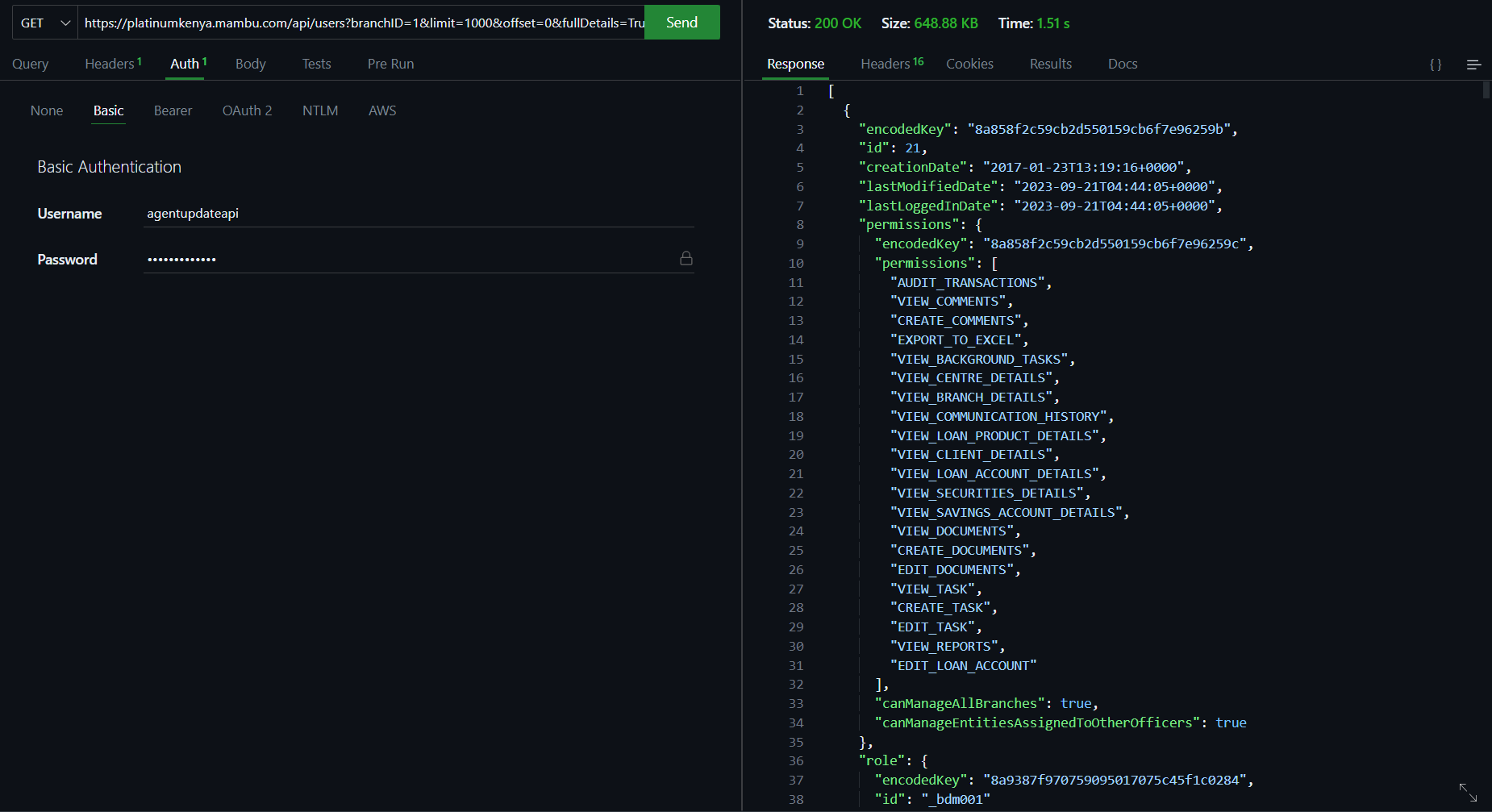
***Figure 9 - Querying Mambu using the bulkposting User(Platinum Uganda)***

**

***Figure 10 - Querying Mambu using the journalapi User(Momentum)***

**

***Figure 11 - Querying Mambu using the agentupdateapi User(Momentum)***

**