Phishing Testing Platform Project

Rosha Razmarafar
Maltepe University
SE 356 01 Web Application Development
Assist. Prof. Dr. Emre OLCA
12.1.2024

Phishing Testing Platform

The "Phishing Testing Platform" project is our guide, offering a hands-on adventure to understand and combat the notorious phishing attacks. In this exploration, we build our own phishing testing playground, equipped with tools to recognize, deflect, and learn from these online scams. Our mission involves creating fake emails, selecting targets (all simulated, of course!), and analyzing how people interact with our faux communications.

In the contemporary landscape of digital vulnerabilities, phishing attacks have emerged as formidable threats, necessitating a proactive stance in comprehending and mitigating their impact. The "Phishing Testing Platform" emerges as a proactive initiative, employing ASP.NET and MSSQL Server for their robustness and applicability in constructing a dynamic web application.

The Administration Panel, a pivotal facet, allows for the meticulous creation of deceptive email templates, strategic management of target email addresses, and comprehensive tracking and analysis of email interactions. Concurrently, the Phishing Email Template Creation Page facilitates the meticulous crafting of authentic-looking yet simulated email templates.

This report meticulously delineates the developmental trajectory of the project, encompassing planning, design, database architecture, server-side programming, and rigorous testing. Ethical considerations, an intrinsic aspect of cybersecurity, underscore our commitment to user privacy and responsible data handling. The narrative unfolds as a testament to our dedication to unraveling the intricacies of phishing and fortifying our digital perimeters against evolving cyber threats.

1. Project Components and Descriptions

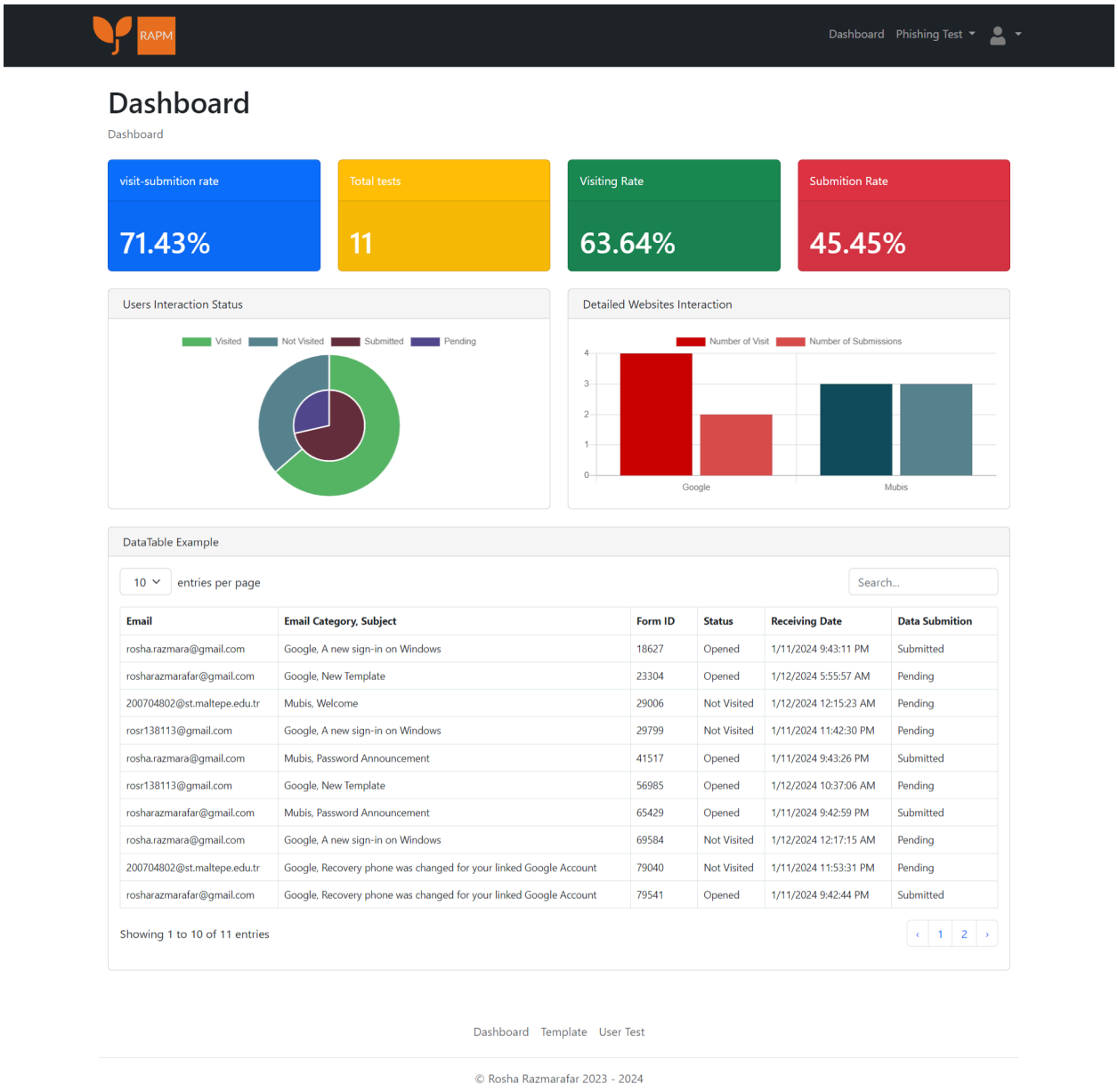
In this project which has been utilizing ASP.NET framework along with MVC library, we have designed multiple component and pages to achieve our goal.

a. Administration Panel (Dashboard):

The Administration Panel serves as the primary interface for system administrators, providing an extensive overview of the platform's activities and essential insights derived from simulated phishing tests on email users. Utilizing libraries such as Chart.js, the panel incorporates bar charts and area charts to visualize user interactions, including submission and click rates and emails staus. A bar chart enables administrators to comprehensively compare interactions across different phishing categories.

Furthermore, the dashboard integrates a DataTable presenting a comprehensive summary of users, their statuses, and their engagement with phishing emails. This real-time tracking mechanism allows administrators to monitor ongoing tests effectively.

This page is connected to an ADO.NET Entity Data Model in order to retrieve the table view from database to bind with the table section in the page. The data table is transferred as a model to view and is updated after each sql operation request is made in order to prevent from having cached or stale data.



b. Phishing Email Template Creation Page:

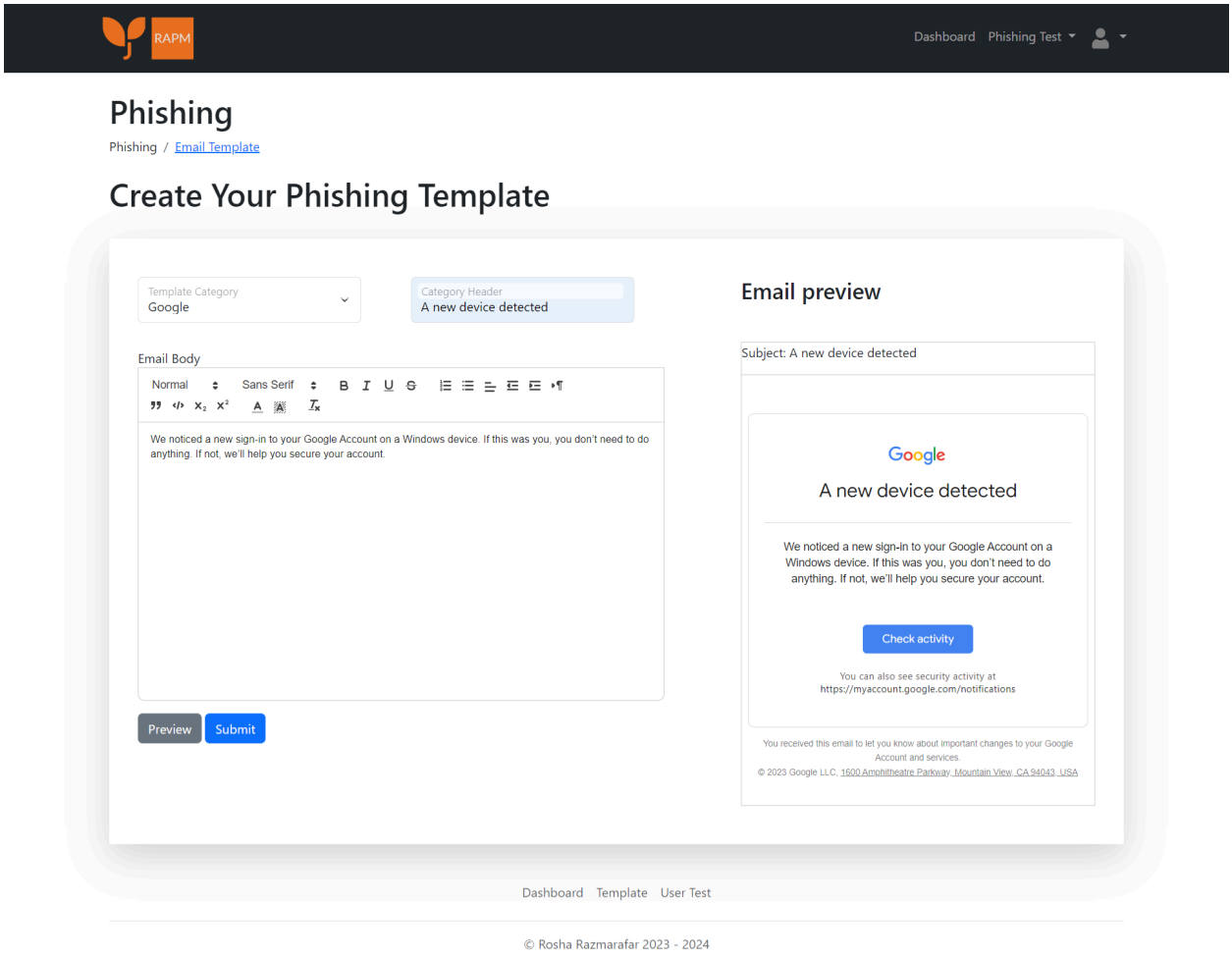
The Phishing Email Template Creation Page functions as a central hub for crafting deceptive yet authentic-looking email templates. By incorporating login pages and other elements, this section simulates scenarios where users may be prompted to disclose sensitive information.

To streamline the template creation process, presaved dropdowns and lists have been implemented. Administrators can choose from three template formats in the Category

dropdown, each featuring a specific HTML header string with a predefined logo and fixed template content, such as Netflix or Google.

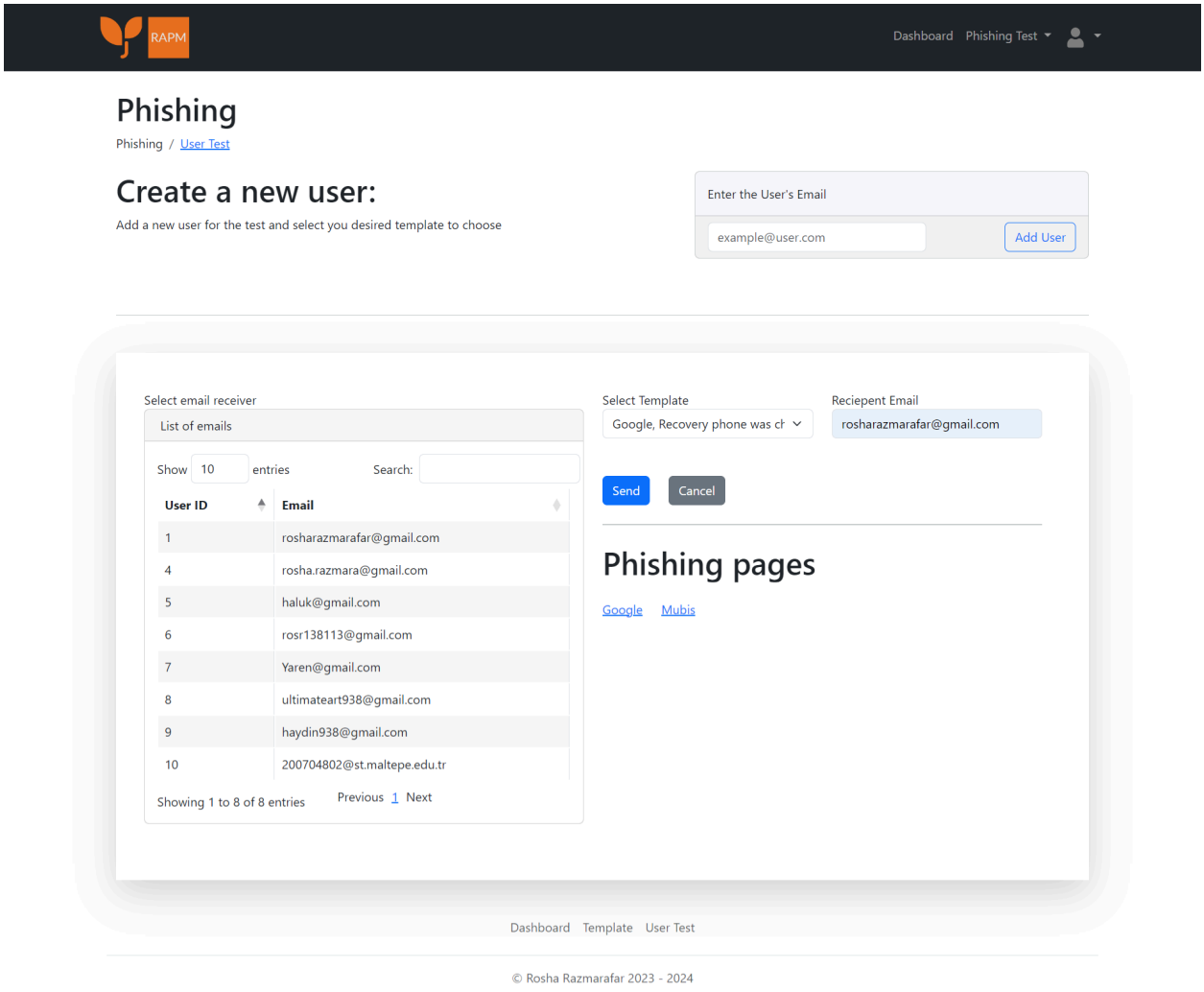
The header section dictates the email subject, allowing administrators to choose from predefined subjects like "Security Alert" or "Failed Payment." A library of text editors facilitates the construction of the email body, including the phishing link. Admins can generate unique links by providing a keyword in a text input field.

The page includes a preview feature enabling administrators to visualize their phishing email before deployment.



c. Determination and Management of Target Email Addresses

In this section, administrators can send phishing emails by selecting targets through a DataTable created with jQuery DataTables CSS and JS components. Administrators can choose from various email templates to deploy, and the page provides three additional links to observe the fake phishing pages associated with the sent emails.

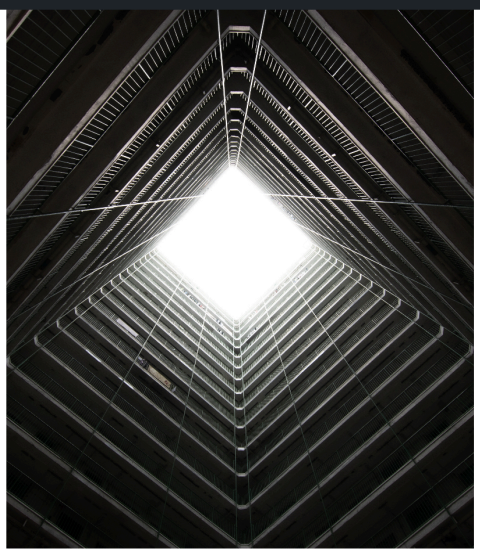


d. Log in and Sign up

To access the administrator panel, a signup page is provided for registration. Upon registration, administrators can log in to their dedicated panel for managing and overseeing phishing tests.

RAPM

Dashboard Phishing Test



Welcome back!

Invalid email or password

Email address
rosharazmarafar@gmail.com

Password

Remember password

SIGN IN

Forgot password?

Not a member yet? Signup

Dashboard Template User Test

RAPM

Dashboard Phishing Test

Registration Form

Full Name

Email address

Password

Repeat Password

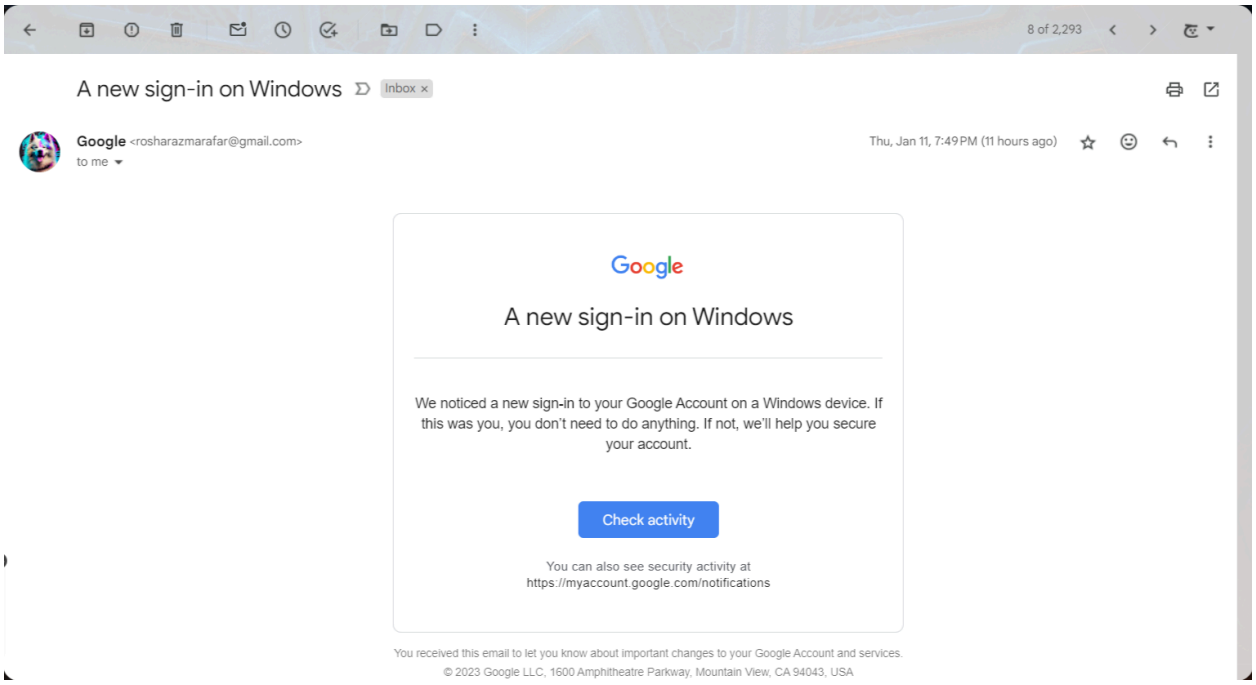
Remember password

SIGN UP

Already a member? Login

Dashboard Template User Test

e. Phishing Emails



The above figure shows the sample of a template test that has been sent to a local email.
The main idea for sending emails was to utilize the MailKit library within a .NET framework for implementing email sending functionality. In the "SendEmail" HTTP POST action method, user input, including the selected email template and recipient's email address, is processed. The code establishes an asynchronous connection to the SMTP server, conducts secure authentication using sender credentials, and transmits the email using the SendAsync method.
However they main issue for sending a traceable email is the link generation for the buttons.
To overcome this issue, in the attached links, special variables are also being attached such as user id and a form id which is a random unique variable to be able to connect this to the link manually.

```
https://localhost:44326/PhishingEmails/Googletemp.aspx?user_id={{USER_ID}}
&form_id={{Form_Id}}
```

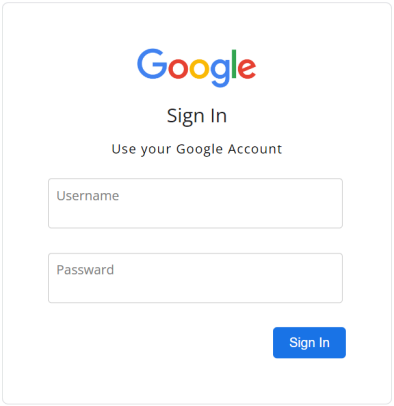
In this sample link, there is an attachment that likely contains important information or additional content. The process involves retrieving the body of an email, and during this retrieval, placeholders such as {{USER_ID}} and {{Form_Id}} are utilized. These placeholders serve as dynamic markers that will later be replaced with their corresponding values. Once replaced, the modified email content is inserted into a database. This method proves to be particularly valuable for tracking emails, especially when coupled with unique form IDs. By incorporating these special form IDs, organizations can effectively monitor and trace the emails they send, allowing for comprehensive tracking and analysis.

f. Fake Phishing Pages

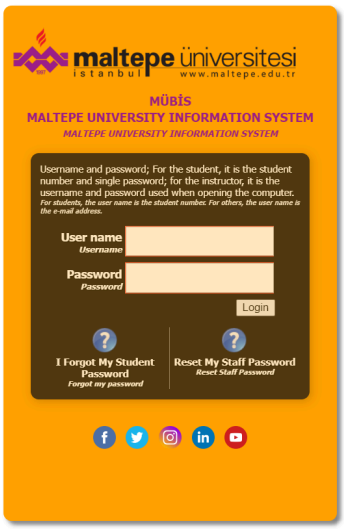
These pages serve as destination pages for phishing emails, redirecting users to cloned versions of legitimate websites such as Mubis, and Google. These pages are meticulously designed to be convincing, ensuring their effectiveness in phishing tests.

These pages are being developed and constructed on ASPX Web Forms in order to get direct related page load functions to record the interactions easier and faster. While the page is loaded the full link is being retrieved from server and is interpreted to extract the form Id and user ID of the email. Upon obtaining the formID the interactions of user is being recorded into database

Google Clone Page:



Mubis Clone Page:



2. Database

For this project, a constructed and architectured database was needed to gather the relation of targets, emails, templates and their interactions. The database, designed using Microsoft SQL Server, serves as the backbone for storing and managing essential information related to phishing simulations.

- The Admins table is independent and serves to authenticate administrators.
- The PhishingEmails table is related to Received Emails and contains details about phishing email templates that are made.
- The Users table stores unique user information.
- The ReceivedEmails table connects users, emails, and the date of receipt and the form id of their specific link along with the email status.
- The PhishingInteractions table captures interactions between users and phishing websites, tracking clicks and submissions.
- Since the FormID is also a unique variable it can be used to retrieve many data from mostly all tables.

a. Table Relations:
   i. **One-to-Many Relationships:**
   Users to ReceivedEmails:
      - A user can receive multiple emails.
      - Each received email is associated with one and only one user.
      - Relationship: One-to-Many (via the UserID in the ReceivedEmails table)

This means that a single user can receive multiple phishing emails, and each received email is linked to one user.

PhishingEmails to ReceivedEmails:

- Multiple users may receive the same phishing email.
- Each received email is associated with one and only one phishing email.
- Relationship: One-to-Many (via the EmailID in the ReceivedEmails table)

This indicates that a phishing email can be sent to multiple users, and each received email is linked to one specific phishing email.

PhishingWebsites to PhishingInteractions:

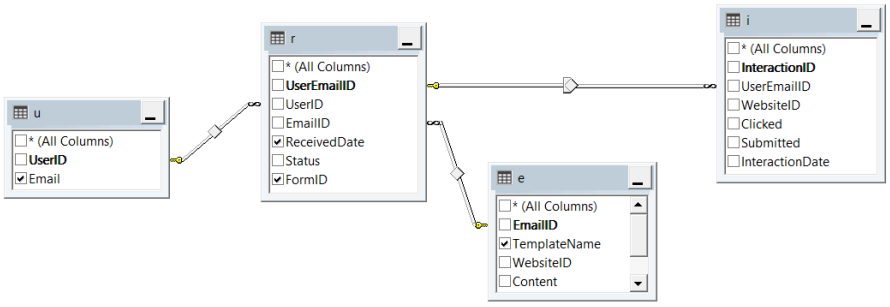
- A phishing website can have multiple interactions.
- An interaction is associated with one and only one phishing website.
- Relationship: One-to-Many (via the WebsiteID in the PhishingInteractions table)

Users to PhishingInteractions:

- A user can have multiple interactions.
- An interaction is associated with one and only one user.
- Relationship: One-to-Many (via the UserEmailID in the PhishingInteractions table)

Admin table is an independent table in this database since it only holds Admins details.

In the project, a saved view serves as a pivotal component to streamline and simplify the process of obtaining data for the dashboard. This view is designed to encapsulate specific data by orchestrating joins and establishing essential connections between the required tables. By doing so, it creates a comprehensive and cohesive representation that acts as a predefined dynamic table. This strategic approach alleviates the need for further operations or complex queries when extracting information for the dashboard. Essentially, the saved view acts as a preconfigured snapshot of interconnected data, enabling efficient retrieval and utilization in the dashboard without the necessity for additional computations or manual interventions. This not only enhances the overall performance but also contributes to the maintainability and scalability of the system by centralizing the logic for data extraction.



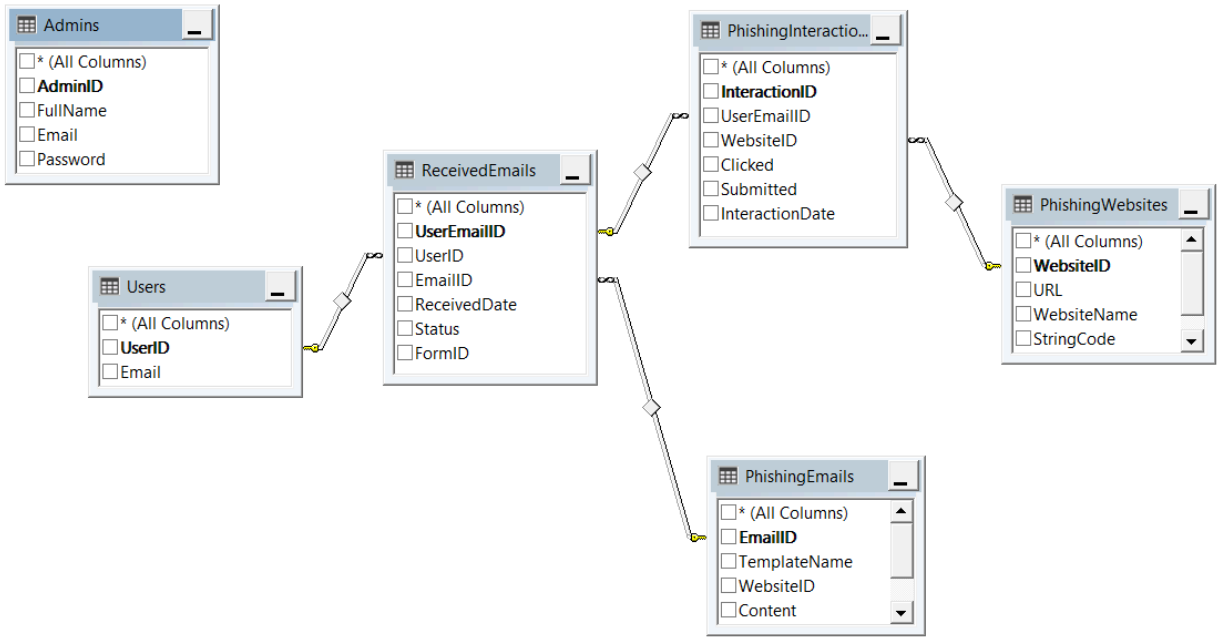
Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...	Or...	Or...

```
SELECT DISTINCT
                r.FormID, u.Email AS UserEmail, e.TemplateName AS EmailCategorySubject, CASE WHEN i.Clicked = 1 THEN 'Opened' WHEN i.InteractionID IS NULL THEN 'Not Visited' ELSE 'Pending' END AS Status,
                r.ReceivedDate AS ReceivingDate, CASE WHEN i.Submitted = 1 THEN 'Submitted' ELSE 'Pending' END AS DataSubmission
FROM        dbo.ReceivedEmails AS r LEFT OUTER JOIN
                dbo.PhishingInteractions AS i ON r.UserEmailID = i.UserEmailID INNER JOIN
                dbo.PhishingEmails AS e ON r.EmailID = e.EmailID INNER JOIN
                dbo.Users AS u ON r.UserID = u.UserID
```

FormID	UserEmail	EmailCategorySubject	Status	ReceivingDate	DataSubmission
18627	rosha.razmara@gmail.com	Google, A new sign-in on Wind...	Opened	2024-01-11 21:43:11.253	Submitted
29006	200704802@st.maltepe.edu.tr	Mubis, Welcome	Not Visited	2024-01-12 00:15:23.030	Pending
29799	rosr138113@gmail.com	Google, A new sign-in on Wind...	Not Visited	2024-01-11 23:42:30.787	Pending
41517	rosha.razmara@gmail.com	Mubis, Password Announcement	Opened	2024-01-11 21:43:26.533	Submitted
65429	rosharazmarafar@gmail.com	Mubis, Password Announcement	Opened	2024-01-11 21:42:59.733	Submitted
69584	rosha.razmara@gmail.com	Google, A new sign-in on Wind...	Not Visited	2024-01-12 00:17:15.557	Pending
79040	200704802@st.maltepe.edu.tr	Google, Recovery phone was ch...	Not Visited	2024-01-11 23:53:31.673	Pending
79541	rosharazmarafar@gmail.com	Google, Recovery phone was ch...	Opened	2024-01-11 21:42:44.807	Submitted

1 of 9 | Cell is Read Only.

b. Database Schema:



3. References and resources used:
    I. https://startbootstrap.com/snippets/sign-in-split
    II. https://quilljs.com/docs/quickstart/
    III. https://mubis.maltepe.edu.tr/KullaniciGirisi.aspx
    IV. https://getbootstrap.com/docs/5.3/getting-started/introduction/
    V. https://datatables.net/
    VI. https://www.chartjs.org/docs/latest/getting-started/
    VII. https://morioh.com/a/d1d6768b9683/netflix-login-page-clone-with-html-and-css
    VIII. https://startbootstrap.com/template/sb-admin