

FACULTY OF INFORMATION MANAGEMENT

Software Engineering

**ADVENTIST UNIVERSITY OF CENTRAL AFRICA (AUCA)**

**RWANDA DRIVING LICENSE MANAGEMENT SYSTEM**

**INSY 8314**

**Web Design Project**

**Submitted By**

**NDAYISABA KAMARIZA Belie**

**ID: 27174**

**May 2025**

**Table Of Content**

**[System Overview 3](#_Toc32282)**

[Technologies Used 3](#_Toc1996)

[Tools Utilized 3](#_Toc9491)

[Platform 3](#_Toc31063)

[System Role 3](#_Toc9906)

**[Introduction 4](#_Toc21729)**

**[Problem Statement 4](#_Toc18819)**

**[Objective 4](#_Toc18491)**

**[Entity-Relationship Diagram (ERD) 5](#_Toc26539)**

**[Sign-Up Page 5](#_Toc25158)**

**[Functional Flow 6](#_Toc9382)**

**[Login Page 7](#_Toc20516)**

[Functional Flow 7](#_Toc6718)

**[Home Page 8](#_Toc10466)**

[Functional Flow 8](#_Toc32274)

**[Candidate Registration 9](#_Toc25437)**

[Functional Flow 10](#_Toc3161)

**[Candidate Overview 11](#_Toc12408)**

[Functional Flow 11](#_Toc10704)

[Candidate Update 12](#_Toc990)

[Functional Flow 12](#_Toc32233)

[Candidate Deletion 13](#_Toc10202)

[Functional Flow 14](#_Toc4029)

**[Application Registration 15](#_Toc23837)**

[Functional Flow 15](#_Toc853)

**[Application Overview 16](#_Toc19802)**

[Functional Flow 16](#_Toc16945)

[Application Update 17](#_Toc4134)

[Functional Flow 17](#_Toc29002)

[Application Deletion 18](#_Toc15234)

[Functional Flow 18](#_Toc26591)

**[Grade Registration 19](#_Toc1518)**

[Functional Flow 19](#_Toc13758)

**[Grade Overview 21](#_Toc28165)**

[Functional Flow 21](#_Toc31996)

[Grade Deletion 22](#_Toc15332)

[Functional Flow 22](#_Toc16943)

[Grade Update 23](#_Toc4852)

[Functional Flow 23](#_Toc12196)

**[License Registration 24](#_Toc16074)**

[Functional Flow 24](#_Toc27436)

**[License Overview 26](#_Toc31797)**

[Functional Flow 26](#_Toc23417)

[License Update 27](#_Toc18371)

[Functional Flow 27](#_Toc24855)

[License Deletion 28](#_Toc17714)

[Functional Flow 29](#_Toc26765)

**[Candidate Report 29](#_Toc12205)**

[Functional Flow 29](#_Toc9769)

**[Candidate Report Display 30](#_Toc206)**

[Functional Flow 31](#_Toc4482)

**[Application Report 32](#_Toc11440)**

[Functional Flow 32](#_Toc9654)

**[Application Report Display 33](#_Toc925)**

[Functional Flow 33](#_Toc21743)

**[Grade Report 34](#_Toc3607)**

[Functional Flow 35](#_Toc20659)

**[Grade Decision Report Display 36](#_Toc7764)**

[Functional Flow 36](#_Toc5638)

**[License Report 37](#_Toc27231)**

[Functional Flow 37](#_Toc25461)

**[License Report Display 38](#_Toc23678)**

[Functional Flow 38](#_Toc12404)

**[Conclusion 40](#_Toc29582)**

# System Overview

The Rwanda Driving License Management System is a web-based platform developed to simplify the application, testing, and issuance process for driving licenses in Rwanda. This system uses modern web technologies and runs in a local server environment for development and testing.

## Technologies Used

|  |  |
| --- | --- |
| Layer | Technology |
| Frontend | HTML5, CSS3,JavaScript |
| Backend | PHP |
| Database | MySQL |

## Tools Utilized

* XAMPP – Provides Apache, PHP, and MySQL for local hosting and testing.
* Visual Studio Code (VS Code) – Source code editor used for writing and debugging PHP, HTML, CSS and JavaScript.
* phpMyAdmin – Browser-based tool for managing MySQL databases.

## Platform

* Execution Environment: Localhost (offline testing using XAMPP)
* Supported Browsers: Google Chrome, Opera, Microsoft Edge

## System Role

|  |  |
| --- | --- |
| Role | Privileges & Responsibilities |
| Admin | The system includes one primary role the administrator  who can: - Register and manage candidate records - Handle application submissions - Record theory and practical test scores - Approve or reject license applications - Issue licenses and generate reports |

# Introduction

The **Rwanda Driving License Management System** is a web-based application developed to streamline the process of applying for, managing, and issuing driving licenses in Rwanda. The system replaces manual, paper-based workflows with a digital platform that is more efficient, transparent, and user-friendly.

Designed specifically for use by government administrators, this system allows for candidate registration, application tracking, test grade management, and license issuance all from a centralized interface. The aim is to reduce delays, minimize errors, and ensure secure handling of license-related processes.

# Problem Statement

In Rwanda, the traditional process of applying for and managing driving licenses is largely manual, involving physical paperwork, in person visits to government offices, and delays in communication between applicants and officials. This system is time-consuming, prone to human error, and lacks transparency in tracking application status, test results, and license issuance.

As the number of applicants grows, administrative staff become overwhelmed, leading to inefficiencies, mismanagement of records, and longer processing times. There is also no centralized way to validate candidate eligibility or enforce requirements such as passing grades or having a provisional license.

To address these challenges, there is a need for an automated, web-based system that simplifies and secures the driving license workflow from candidate registration to license issuance while enabling government officials to manage operations efficiently.

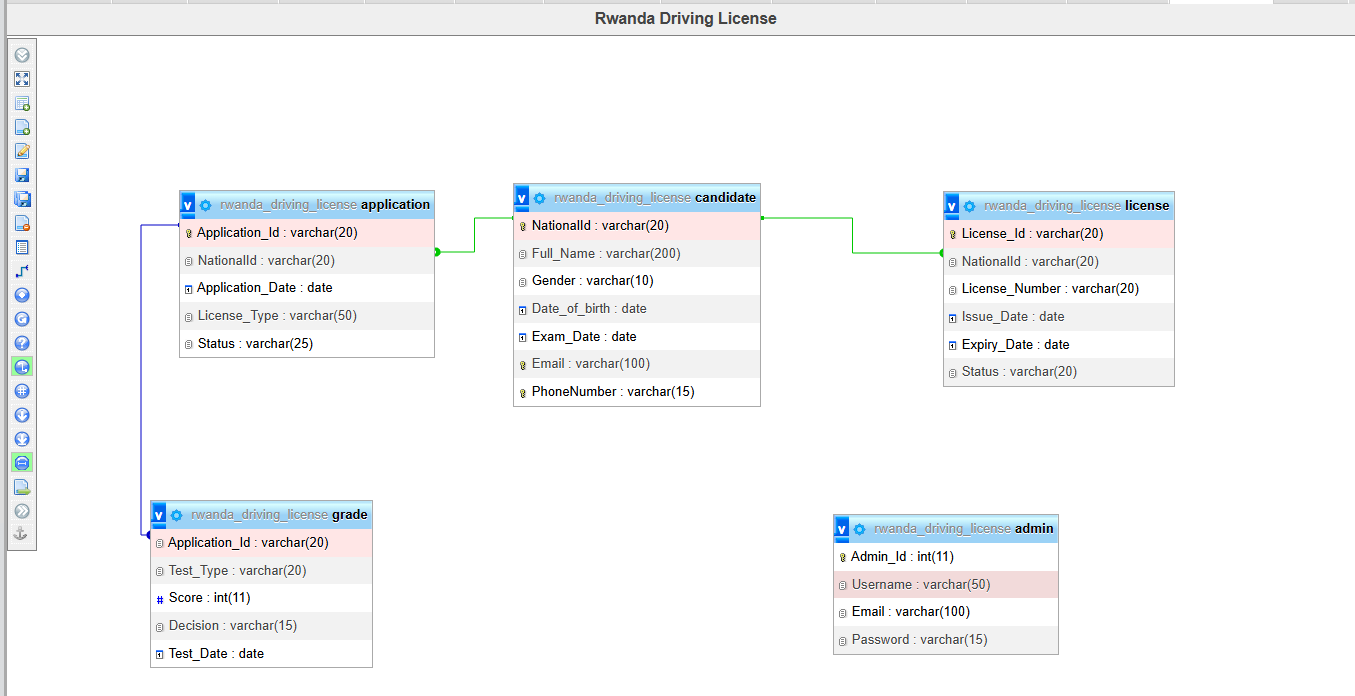
# Objectives

* To **digitize** the driving license application and management process in Rwanda.
* To enable **online scheduling** of theory and practical driving tests.
* To **record and manage** candidates grades in a secure and organized way.
* To **speed up** the approval and issuance of driving licenses.
* To **reduce paperwork** and physical visits to government offices.
* To improve **transparency and communication** between candidates and administrators.
* To provide administrators with an **easy-to-use system** for reviewing applications, managing tests, and issuing licenses.
* To ensure the **security and privacy** of user data throughout the system.

Build a comprehensive digital solution for the Rwandan driving license system, enabling citizens to apply for licenses, check their application status, and access issued licenses online, while providing authorities with administrative tools for reviewing, approving, and maintaining license records.

# Entity-Relationship Diagram (ERD)

The Entity-Relationship Diagram (ERD) illustrates the logical structure of the Rwanda Driving License Management System database. It defines the main entities involved such as **Candidate**, **Application**, **Grade**, **License**, and **Admin** and outlines how these entities are related.



# Sign-Up Page

Sign-up page is designed to enable new users to create accounts effortlessly, initiating their journey to obtaining a driving license. The system aims to streamline the application process by allowing online registration, progress tracking, and confidence-building for users. It targets individuals in Rwanda seeking a driving license, providing a simple and smooth interface as emphasized by the tagline: "We are here to make your journey to becoming a licensed driver simple and smooth. Apply online, track your progress, and get ready to hit the road with confidence!"

# Functional Flow

1. User Access: The user navigates to the sign-up page, greeted by a welcome message and a form.

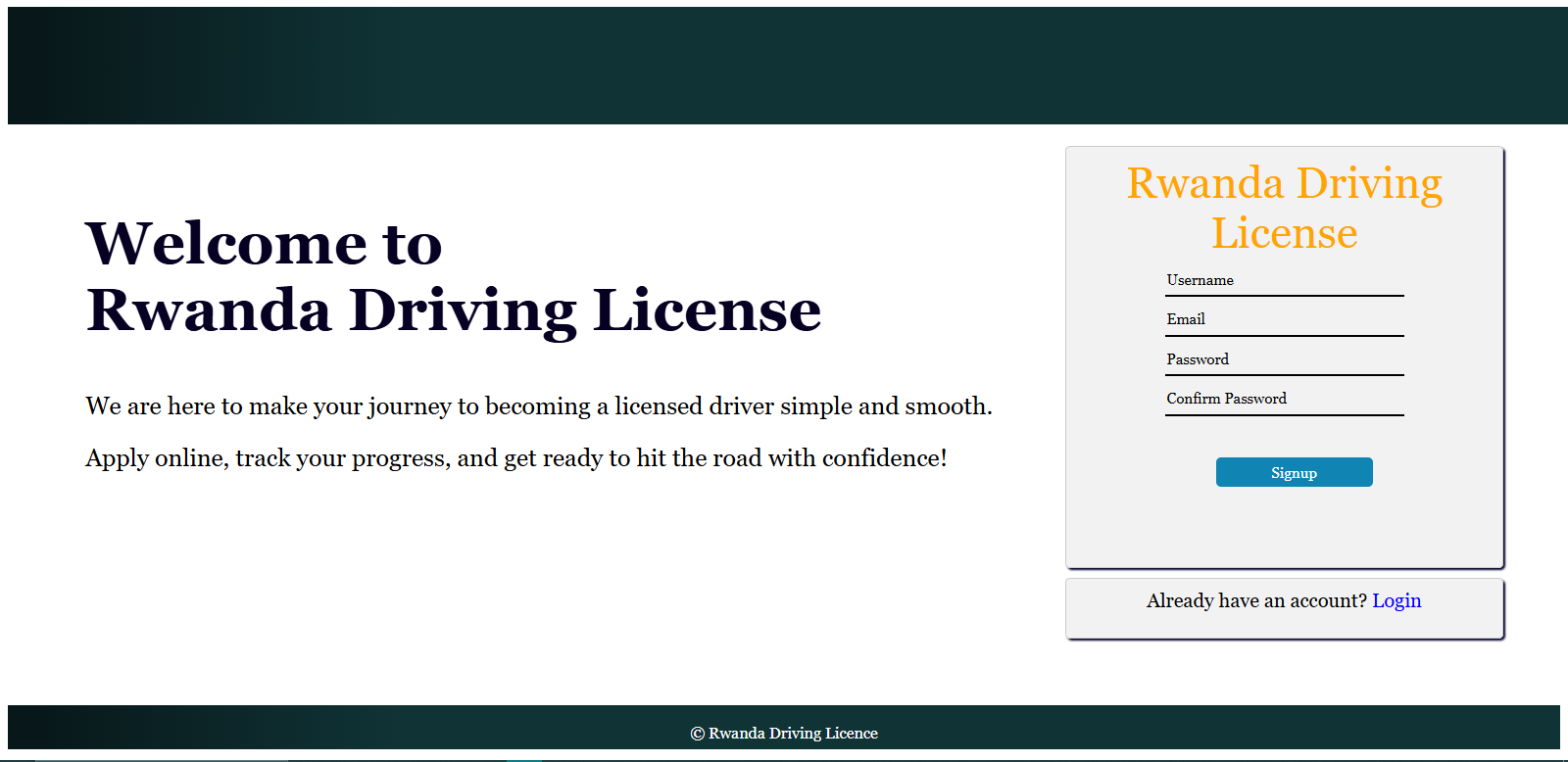
2. Form Submission: The user enters Username, Email, Password, and Confirm Password, then clicks "Sign-up." Client-side validation ensures all fields are filled.

3. Back-end Processing:

PHP connects to the "Rwanda\_driving\_license" MySQL database.

* Checks if the password matches the confirmation.
* Queries the "admin" table to verify if the username or email already exists.
* If unique and passwords match:
* Inserts the user data (Email, Username, Password) into the "admin" table.
* Displays a "Successfully" alert and redirects to "login.php."
* If validation fails:Displays alerts (e.g., "Incorrect Password," "User Already Exist," or "Email Already Exist").

1. User Options:Existing users can click "Login" to access their accounts.



Sign-up page offers a basic yet functional platform for user registration, supported by a simple HTML/CSS interface and PHP back-end with MySQL integration. It successfully handles account creation and basic validation, aligning with its purpose of simplifying the licensing process. However, security risks (e.g., plain text passwords, SQL injection vulnerabilities) and limited user experience features (e.g., no password recovery) suggest areas for improvement to enhance reliability and user trust.

# Login Page

Login page enables existing users to access their accounts to manage their driving license applications and track progress. It supports the broader goal of simplifying the licensing process by providing a secure entry point for registered users, aligning with the tagline: "We are here to make your journey to becoming a licensed driver simple and smooth. Apply online, track your progress, and get ready to hit the road with confidence!" The page targets registered users in Rwanda seeking to continue or monitor their licensing journey.

## Functional Flow

1. User Access:The user navigates to the login page, greeted by a welcome message and a login form.

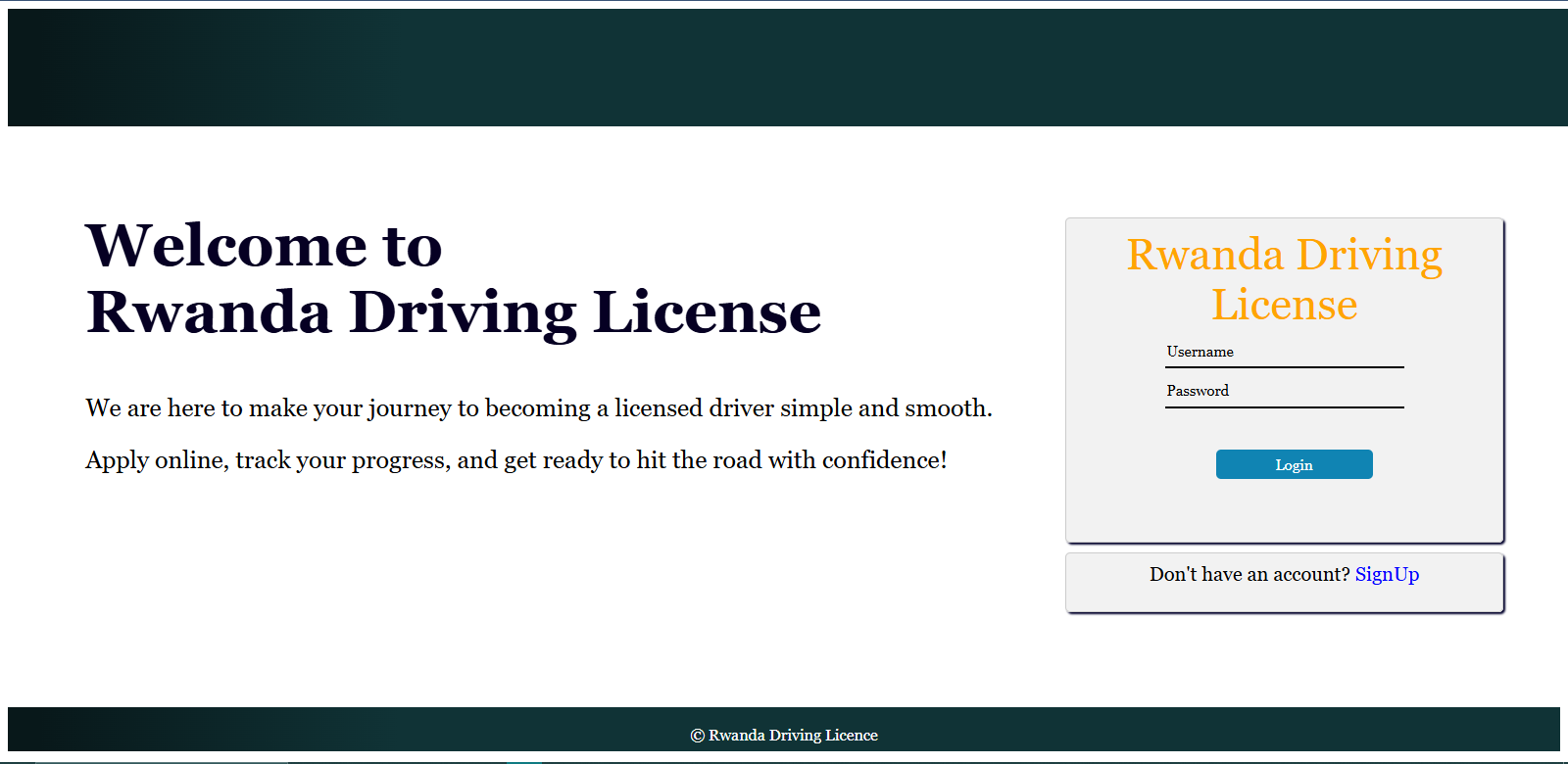
2. Form Submission:

* The user enters Username and Password, then clicks "Login."
* Client-side validation ensures all fields are filled.

3. Back-end Processing:

* PHP connects to the "Rwanda\_driving\_license" MySQL database.
* Executes a query to match the Username and Password against the "admin" table.
* If a match is found (one row returned):
* Starts a session and stores the Username in `$\_SESSION['login']`.
* Redirects the user to "index.php."
* If no match is found:
* Displays an alert ("Account does not Exit!! Create New Account") and redirects to "signup.php."

1. User Options:Users without accounts can click "SignUp" to register via a link to "Signup.php."



Login page provides a functional interface for existing users to access their accounts, supported by a simple HTML/CSS design and PHP back-end with MySQL integration. It effectively authenticates users and redirects them to the main interface or prompts new registrations. However, security concerns (e.g., plain text password storage, lack of prepared statements for SQL injection prevention) and limited error handling suggest the need for enhancements to ensure a robust and secure user experience.

# Home Page

Main dashboard serves as the central hub for authenticated users to manage their licensing journey. It provides access to various sections such as Candidate registration, Application tracking, Grade management, License issuance, and Reports. The page personalizes the user experience by displaying the logged-in user's name and aligns with the system's goal of simplifying the process, as reflected in the welcoming message: "Welcome To RWANDA Driving License."

## Functional Flow

1. User Authentication:

* The page checks for an active session (`$\_SESSION['login']`).
* If no session exists, redirects the user to "login.php."

2. Page Load:

Displays a navigation bar with links to:

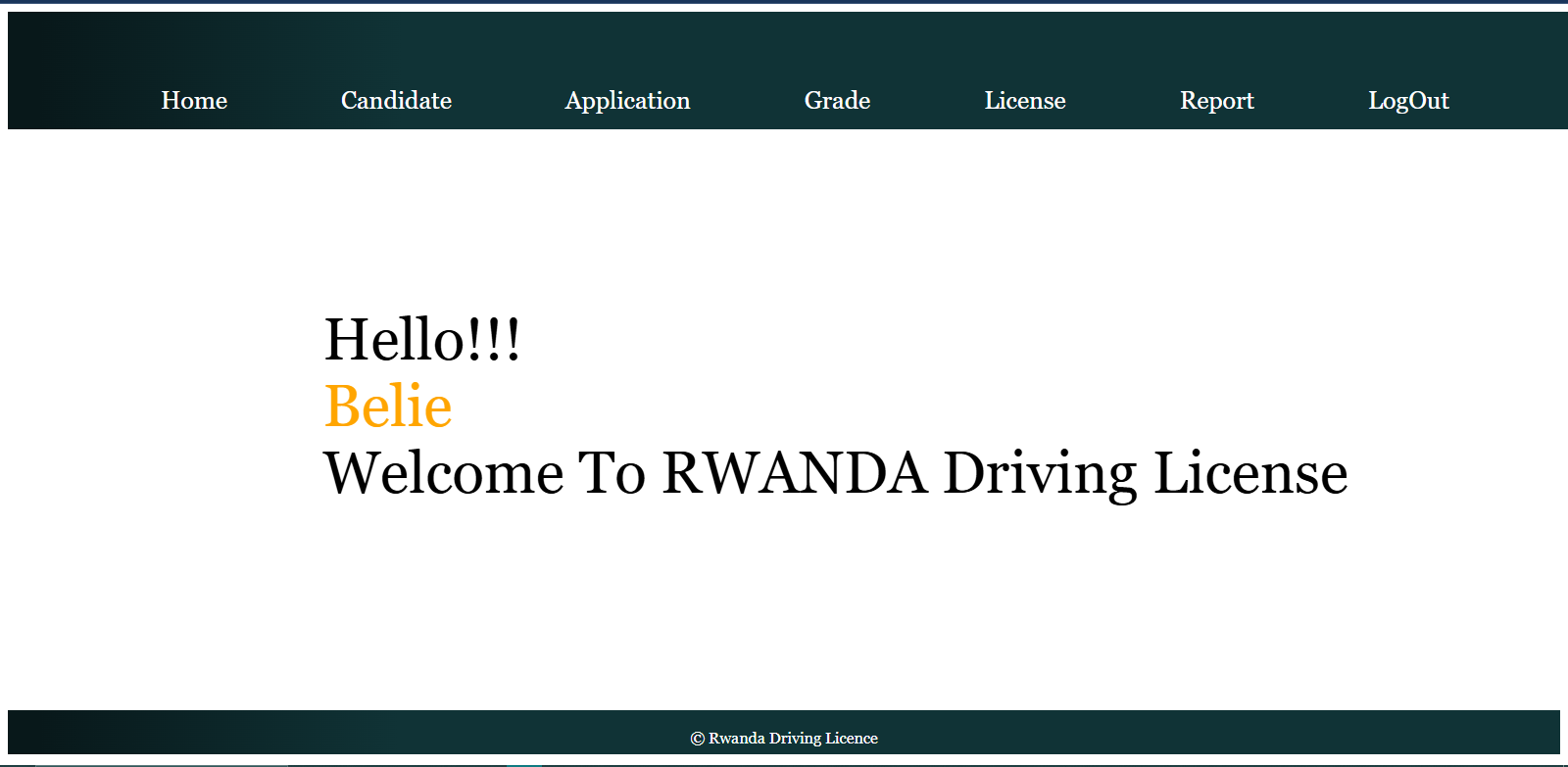
* Home, Candidate (with Register and Overview sub-menus), Application (with Register and Overview sub-menus), Grade (with Register and Overview sub-menus), License (with Register and Overview sub-menus), Report (with Candidate, Application, Decision, and License sub-menus), LogOut

Shows a personalized greeting with the user's username (e.g., "Hello!!! Belie") and a welcome message.

3. User Interaction:

* Users can navigate to any section by clicking the respective links or sub-menus.
* Clicking "LogOut" ends the session and likely redirects to the login page (handled by "Logout.php").

1. Exit:The session persists until the user logs out or the session expires.



Main dashboard offers a user-friendly interface for authenticated users to access key licensing features, supported by HTML and PHP with session management. It effectively organizes navigation with a drop-down menu system and personalizes the experience with the user's name. However, the lack of detailed content on the page and potential security concerns (e.g., session handling without additional validation) suggest opportunities for enhancement to improve functionality and security.

# Candidate Registration

Candidate Registration page allows authenticated users to register as candidates for a driving license. It collects essential personal information (e.g., National ID, Full Name, Gender, Birth Date, Exam Date, Email, Phone Number) to facilitate the licensing process. The page aims to ensure accurate and unique candidate data while validating age eligibility (minimum 18 years), aligning with the system's goal of simplifying and streamlining the journey to becoming a licensed driver.

## Functional Flow

1. User Authentication: Checks for an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

2. Page Load:

* Displays a navigation bar with links to Home, Candidate, Application, Grade, License, Report, and LogOut, including sub-menus for detailed actions.
* Presents a form titled "Candidate Registration" with fields: NationalID, Full Name, Gender (drop-down), Birth Date, Exam Date, Email, and Phone Number.

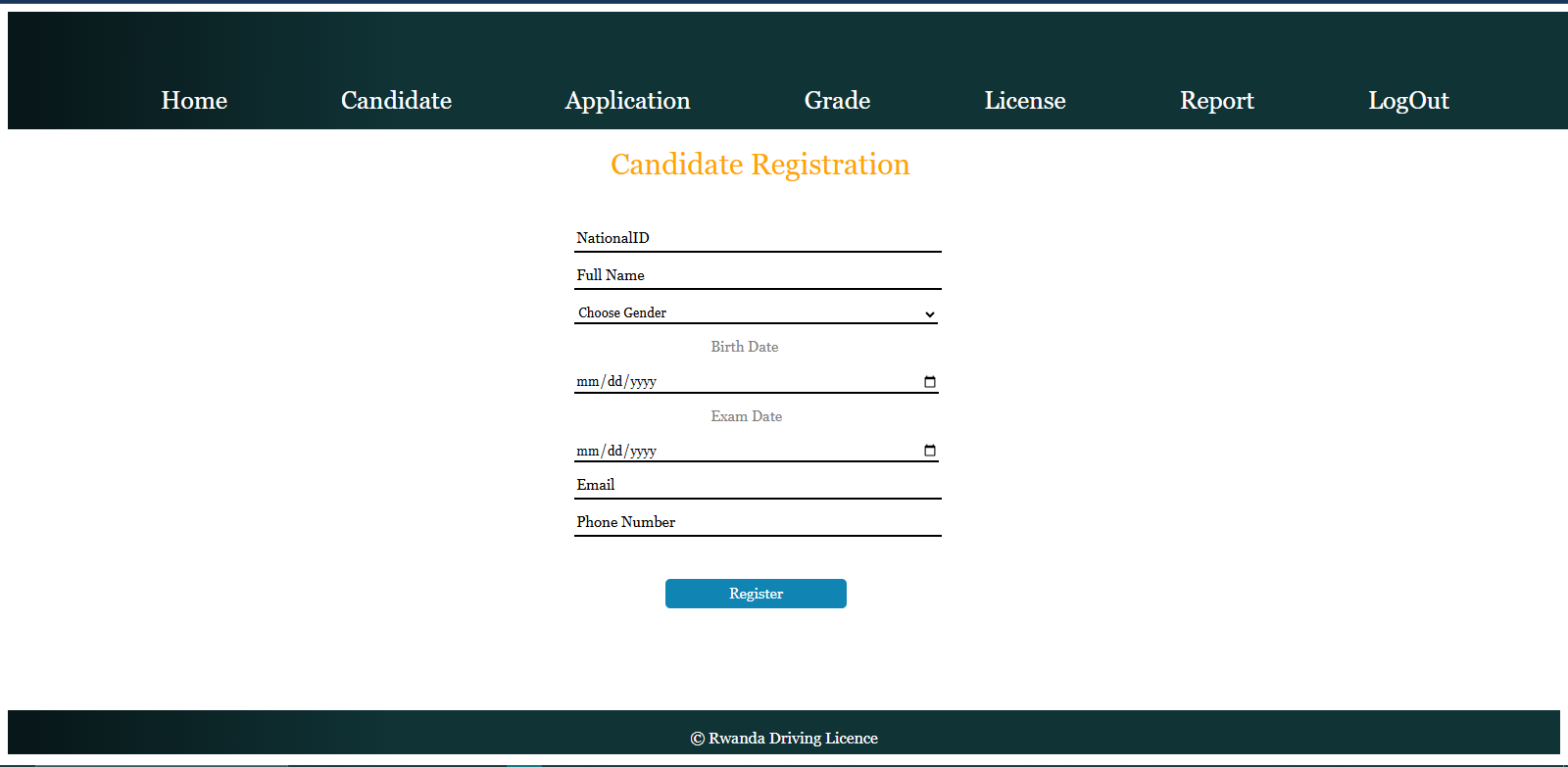
1. Form Submission:User fills and submits the form by clicking "Register."

**PHP validates:**

* National ID length (must be 16 characters).
* Phone number length (must be 10 characters).
* Checks for duplicate National ID, Phone Number, and Email in the "Candidate" table.
* Calculates age from Birth Date to Exam Date; ensures the user is at least 18 years old.
* If validation fails, displays alerts (e.g., "National ID must be 16 Characters," "You must be at least 18 years old") and exits.

If validation passes, inserts data into the "Candidate" table and shows a success ("Registration Successful") or failure alert.

1. User Navigation: Users can explore other sections via the navigation bar or log out via "Logout.php."



Candidate Registration page provides a structured interface for candidate enrollment, supported by HTML, CSS, and PHP with MySQL integration. It includes robust validation for data integrity and age eligibility, enhancing the licensing process's reliability. However, security risks (e.g., lack of prepared statements for SQL injection prevention, plain text data storage) and limited form feedback suggest improvements are needed to ensure a secure and user-friendly experience.

# Candidate Overview

Candidate Overview page allows authenticated users to view a list of all registered candidates, displaying their details such as National ID, Full Name, Gender, Birth Date, Email, and Phone Number. It provides options to update or delete candidate records, facilitating efficient management of candidate data. This page supports the system's goal of streamlining the licensing process by offering a centralized view for monitoring and maintaining candidate records.

## Functional Flow

1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

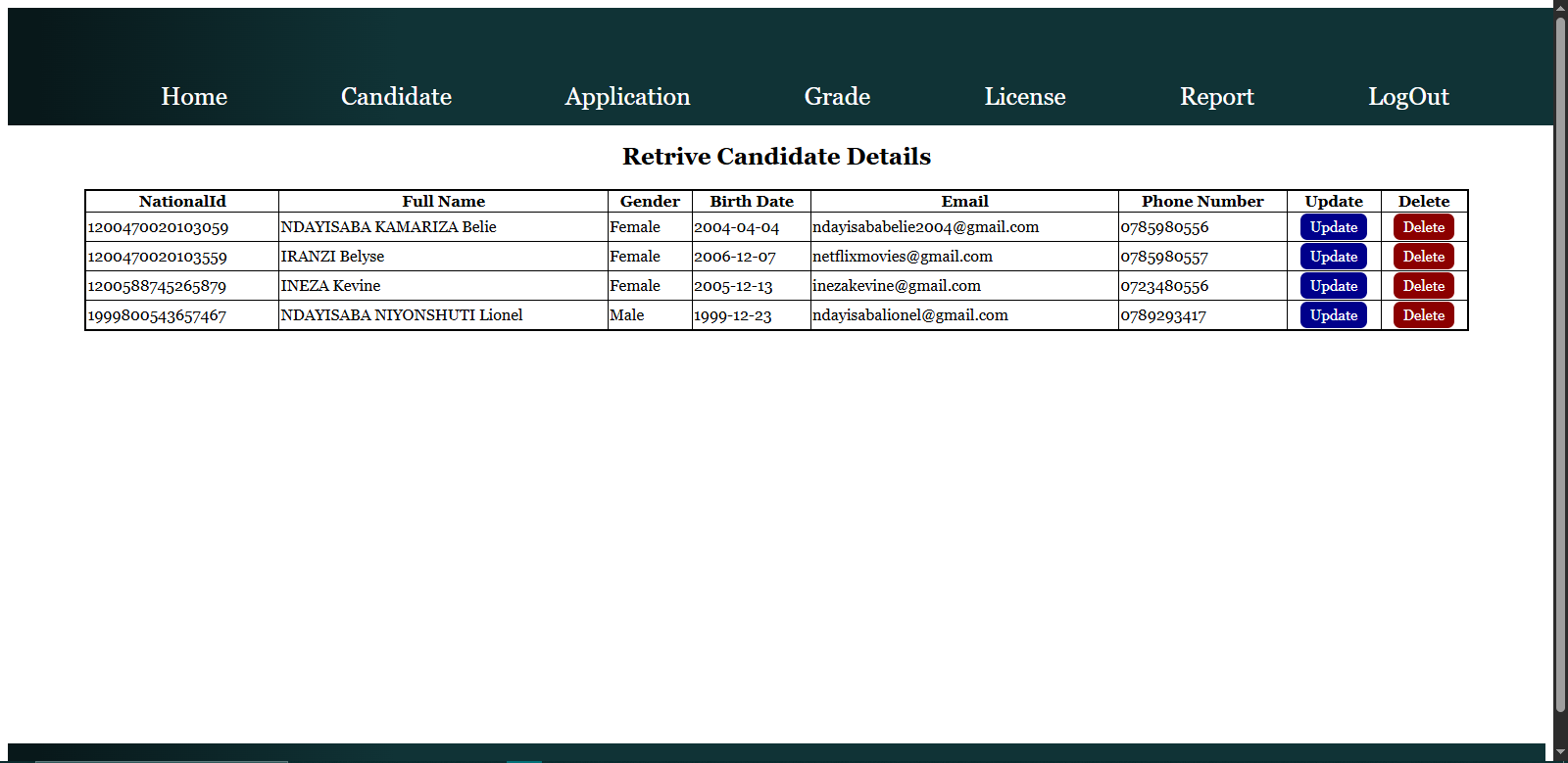
2. Page Load:

* Displays a navigation bar with links to Home, Candidate, Application, Grade, License, Report, and LogOut, including sub-menus for detailed actions.
* Queries the "Candidate" table to retrieve all candidate records.
* Presents the data in a table with columns: NationalId, Full Name, Gender, Birth Date, Email, Phone Number, Update, and Delete.

3. Data Display and Actions:Each row represents a candidate, populated with their details.

* Update Action:A button links to "CandidateUpdate.php" with the National ID as a parameter, prompting a confirmation ("Are Sure You Want to Update?").
* Delete Action:A button links to "Candidatedelete.php" with the National ID as a parameter, prompting a confirmation ("Are Sure You Want to Delete?").

1. User Navigation:Users can navigate to other sections via the navigation bar or log out via "Logout.php.



Candidate Overview page provides a clear and functional interface for viewing and managing candidate records, supported by HTML, CSS, and PHP with MySQL integration. It enables efficient data monitoring with update and delete functionalities, enhancing administrative control. However, security risks (e.g., lack of prepared statements for SQL queries, unencrypted GET parameters) and minimal styling suggest improvements are needed for security and user experience.

## Candidate Update

Candidate Update page allows authenticated users to modify existing candidate records in the licensing system. It retrieves a candidate's details using their National ID (passed via a GET parameter) and enables updates to fields like Full Name, Gender, Birth Date, Exam Date, Email, and Phone Number. The page ensures data accuracy through validation (e.g., age eligibility, unique identifiers) and supports the system's goal of maintaining an efficient and user-friendly licensing process.

## Functional Flow

1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

2. Page Load:

* Displays a navigation bar with links to Home, Candidate, Application, Grade, License, Report, and LogOut, including sub-menus for detailed actions.
* Retrieves the candidate’s data from the "Candidate" table using the National ID and populates the form with existing values.
* Form fields include NationalID (pre-filled, read-only), Full Name, Gender (drop-down), Birth Date, Exam Date, Email, and Phone Number.

3. Form Submission:User updates the fields and clicks "Update."

* **PHP validates:**
* National ID length (must be 16 characters).
* Phone number length (must be 10 characters).
* Age (calculated from Birth Date to Exam Date; must be at least 18 years).
* If validation fails, displays alerts (e.g., "National ID must be 16 Characters," "You must be at least 18 years old") and exits.
* If validation passes, updates the "Candidate" table with new values for Full Name, Gender, Birth Date, Exam Date, Email, and Phone Number.
* Displays a success ("Candidate Update Successfully") or failure alert and redirects to "VCandidate.php" (candidate overview page).

4. User Navigation:Users can navigate to other sections via the navigation bar or log out via "Logout.php."

Candidate Update page provides a straightforward interface for modifying candidate records, supported by HTML, CSS, and PHP with MySQL integration. It ensures data integrity through validation checks and redirects users to the overview page post-update. However, security concerns (e.g., lack of prepared statements for SQL injection prevention, invalidated GET parameter) and limited user feedback suggest enhancements are needed to improve security and usability.

## Candidate Deletion

Candidate Deletion functionality allows authenticated users to remove candidate records from the system using their National ID. It ensures data integrity by checking for dependencies in related tables (e.g., Application, License, Grade) before proceeding with deletion. This feature supports the system's goal of maintaining an accurate and manageable database for the licensing process by preventing orphaned records and ensuring only standalone candidate records are deleted.

## Functional Flow

1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

2. Deletion Request:

* Triggered when a delete request is initiated with a National ID passed via a GET parameter.
* Sanitizes the National ID using `MySQL\_real\_escape\_string` to prevent SQL injection.

3. Dependency Check:

* Queries the "Application," "License," and "Grade" tables to check if the National ID is referenced:
* "Application" table: Checks if the candidate has an application.
* "License" table: Checks if the candidate has a license.
* "Grade" table: Checks if the candidate has grades linked via an application ID.
* If any dependencies are found, displays an alert ("Cannot delete. This ID is in use to other records.") and redirects to "VCandidate.php."

4. Deletion Process:

* If no dependencies exist, deletes the candidate record from the "Candidate" table using the National ID.
* Displays a success ("Deleted successfully") or failure ("Delete failed") alert and redirects to "VCandidate.php."

5. Error Handling:If the delete parameter (`$\_GET['del']`) is not set, displays an alert ("Invalid request").

Candidate Deletion functionality ensures safe removal of candidate records by validating dependencies, supported by PHP and MySQL. It effectively prevents data inconsistencies by blocking deletions if the candidate is linked to other records. However, security risks (e.g., reliance on `MySQL\_real\_escape\_string` alone for SQL injection prevention, unencrypted session handling) and minimal user feedback suggest improvements are needed to enhance security and user experience.

# Application Registration

The Rwanda Driving License Application Registration page enables authenticated users to submit applications for driving licenses. It collects details such as Application ID, National ID, Application Date, and License Type, ensuring candidates follow the required progression (e.g., obtaining a Provisional license before other categories). This feature supports the system's goal of streamlining the licensing process by managing application submissions efficiently and enforcing procedural rules.

## Functional Flow

1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

2. Page Load:

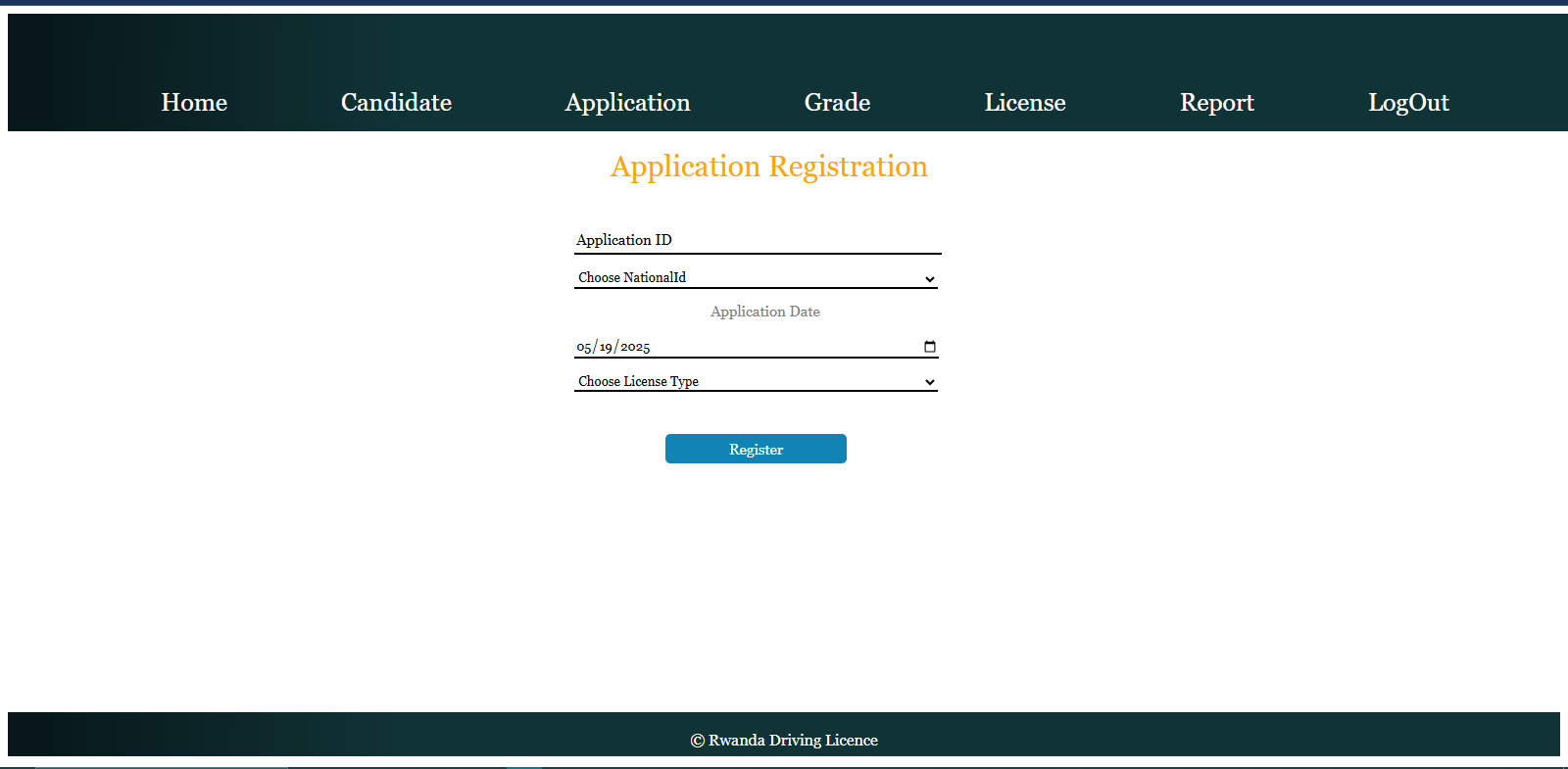
* Displays a navigation bar with links to Home, Candidate, Application, Grade, License, Report, and LogOut, including sub-menus for detailed actions.
* Presents a form titled "Application Registration" with fields: Application ID, National ID (drop-down populated from "Candidate" table), Application Date (pre-filled with today's date), and License Type (drop-down with categories A-F and Provisional).

1. Form Submission:User fills and submits the form by clicking "Register.

**PHP processes:**

* Validates if the requested License Type is not Provisional; checks if the candidate has a Provisional license in the "Application" table.
* If no Provisional license exists for non-Provisional types, displays an alert ("You must first obtain a Provisional License...") and exits.
* Inserts the application data (Application ID, National ID, Application Date, License Type, Status set to "Pending") into the "Application" table.
* Displays a success ("Application Submitted Successfully") or failure ("Application Submission Failed") alert.

1. User Navigation:Users can navigate to other sections via the navigation bar or log out via "Logout.php."



Application Registration page provides a functional interface for submitting license applications, supported by HTML, CSS, and PHP with MySQL integration. It enforces a logical progression by requiring a Provisional license for other categories and includes basic validation, enhancing process integrity. However, security risks (e.g., lack of prepared statements for SQL injection prevention, unencrypted data) and limited validation (e.g., no duplicate Application ID check) suggest improvements are needed for security and reliability.

# Application Overview

Application Overview page allows authenticated users to view a detailed list of all application records, including Application ID, National ID, Full Name, Application Date, License Type, and Status. It provides options to update or delete applications, facilitating effective management of application data. This page supports the system's goal of streamlining the licensing process by offering a centralized view for monitoring and maintaining application records.

## Functional Flow

1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

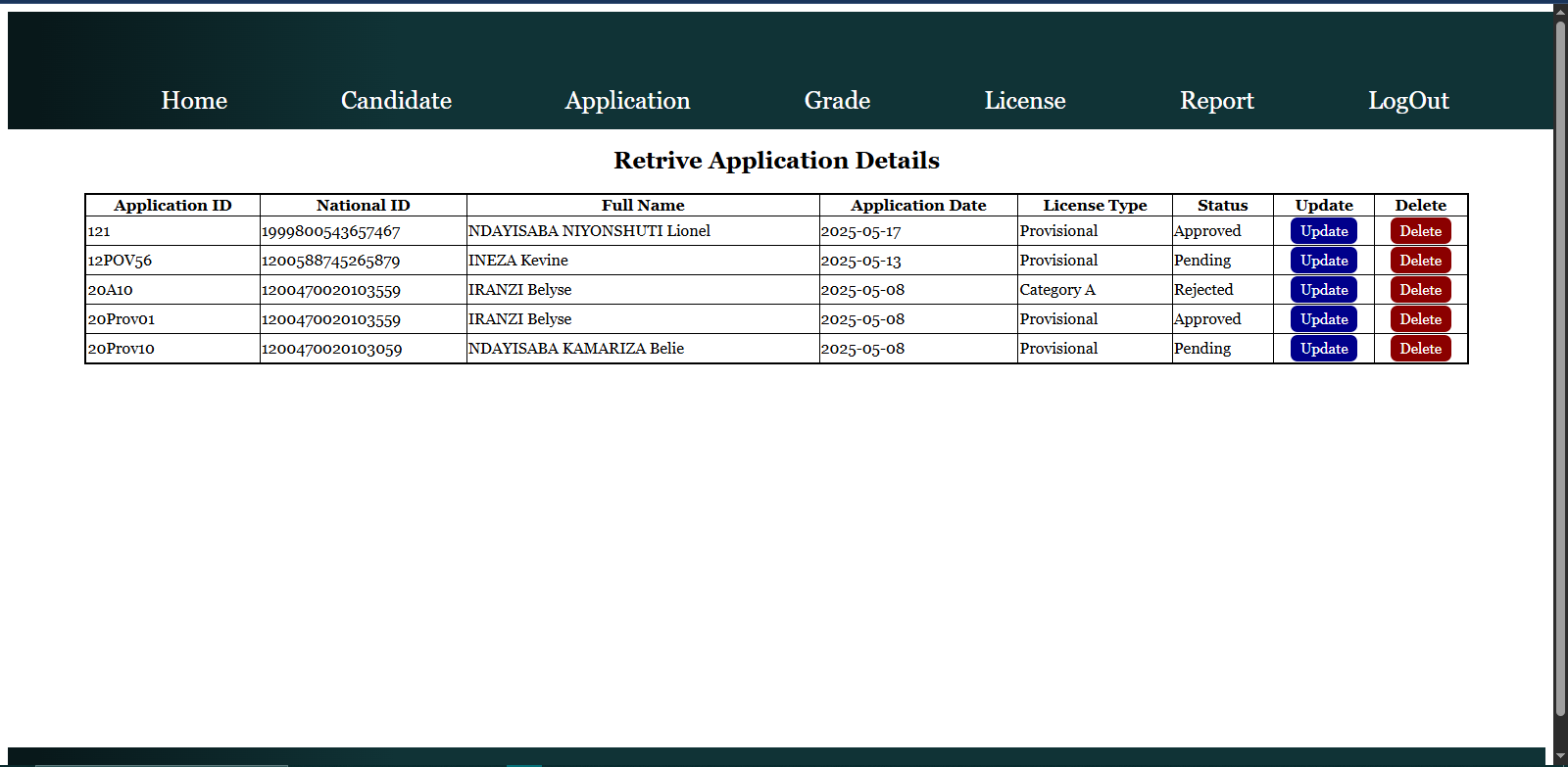
2. Page Load:

* Displays a navigation bar with links to Home, Candidate, Application, Grade, License, Report, and LogOut, including sub-menus for detailed actions.
* Executes a JOIN query on the "Application" and "Candidate" tables to retrieve application details linked with candidate names. Presents the data in a table with columns: Application ID, National ID, Full Name, Application Date, License Type, Status, Update, and Delete.

3. Data Display and Actions:Each row represents an application, populated with its details.

* Update Action:A button links to "ApplicationUpdate.php" with the Application ID as a parameter, prompting a confirmation ("Are Sure You Want to Update?").
* Delete Action: A button links to "Applicationdelete.php" with the Application ID as a parameter , prompting a confirmation ("Are Sure You Want to Delete?").

1. User Navigation:Users can navigate to other sections via the navigation bar or log out via "Logout.php."



Application Overview page provides a clear and functional interface for viewing and managing application records, supported by HTML, CSS, and PHP with MySQL integration. It enhances administrative control with update and delete functionalities, using a JOIN query to enrich data with candidate names. However, security risks (e.g., lack of prepared statements for SQL queries, unencrypted GET parameters) and minimal styling suggest improvements are needed for security and user experience.

## Application Update

Application Update page allows authenticated users to modify existing application records in the system. It retrieves an application using its Application ID (passed via a GET parameter) and enables updates to fields like Application Date, while displaying related details such as National ID. This feature supports the system's goal of maintaining an efficient and accurate licensing process by allowing users to correct or update application information as needed.

## Functional Flow

1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

2. Page Load:

* Displays a navigation bar with links to Home, Candidate, Application, Grade, License, Report, and LogOut, including sub-menus for detailed actions.
* Retrieves the application’s data from the "Application" table using the Application ID and populates the form with existing values.
* Form fields include Application ID (pre-filled, read-only, concatenated with National ID), National ID (pre-filled), and Application Date (editable).

3. Form Submission:User updates the Application Date and clicks "Update."

**PHP processes:**

* Updates the "Application" table with the new Application Date for the specified Application ID.
* Displays a success ("Application Update Successfully") or failure ("Application Update Failed") alert.
* Redirects to "VApplication.php" (application overview page).

4. User Navigation:Users can navigate to other sections via the navigation bar or log out via "Logout.php."

The Rwanda Driving License Application Update page provides a straightforward interface for modifying application records, supported by HTML, CSS, and PHP with MySQL integration. It ensures users can update critical application details efficiently, with automatic redirection to the overview page post-update. However, security risks (e.g., lack of prepared statements for SQL injection prevention, invalidated GET parameter) and minimal validation (e.g., no checks for date consistency) suggest improvements are needed to enhance security and reliability.

## Application Deletion

Application Deletion functionality allows authenticated users to remove application records from the system using their Application ID. It ensures data integrity by checking for dependencies in related tables (e.g., Grade) before proceeding with deletion. This feature supports the system's goal of maintaining an accurate and manageable database for the licensing process by preventing orphaned records and ensuring only standalone application records are deleted.

## Functional Flow

1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

2. Deletion Request:

* Triggered when a delete request is initiated with an Application ID passed via a GET parameter (`$\_GET['del']`).
* Sanitizes the Application ID using `MySQL\_real\_escape\_string` to prevent SQL injection.

3. Dependency Check:

* Queries the "Grade" table to check if the Application ID is referenced.
* If dependencies are found, displays an alert ("Cannot delete. This ID is in use to other records.") and redirects to "VApplication.php."

4. Deletion Process:

* If no dependencies exist, deletes the application record from the "Application" table using the Application ID.
* Displays a success ("Deleted successfully") or failure ("Delete failed") alert and redirects to "VApplication.php."

1. Error Handling:If the delete parameter is not set, displays an alert ("Invalid request").

Application Deletion functionality ensures safe removal of application records by validating dependencies, supported by PHP and MySQL. It effectively prevents data inconsistencies by blocking deletions if the application is linked to grade records. However, security risks (e.g., reliance on `MySQL\_real\_escape\_string` alone for SQL injection prevention, unencrypted session handling) and minimal user feedback suggest improvements are needed to enhance security and user experience.

# Grade Registration

Grade Registration page allows authenticated users to record test results for driving license applicants. It collects details such as Application ID, Test Type, Score, and Test Date, determining a Pass/Fail status based on the score (60-100 for Pass, below 60 for Fail) and updates the corresponding application status (Approved/Rejected). This feature supports the system's goal of streamlining the licensing process by managing test evaluations and application statuses efficiently.

## Functional Flow

1. User Authentication:

- Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

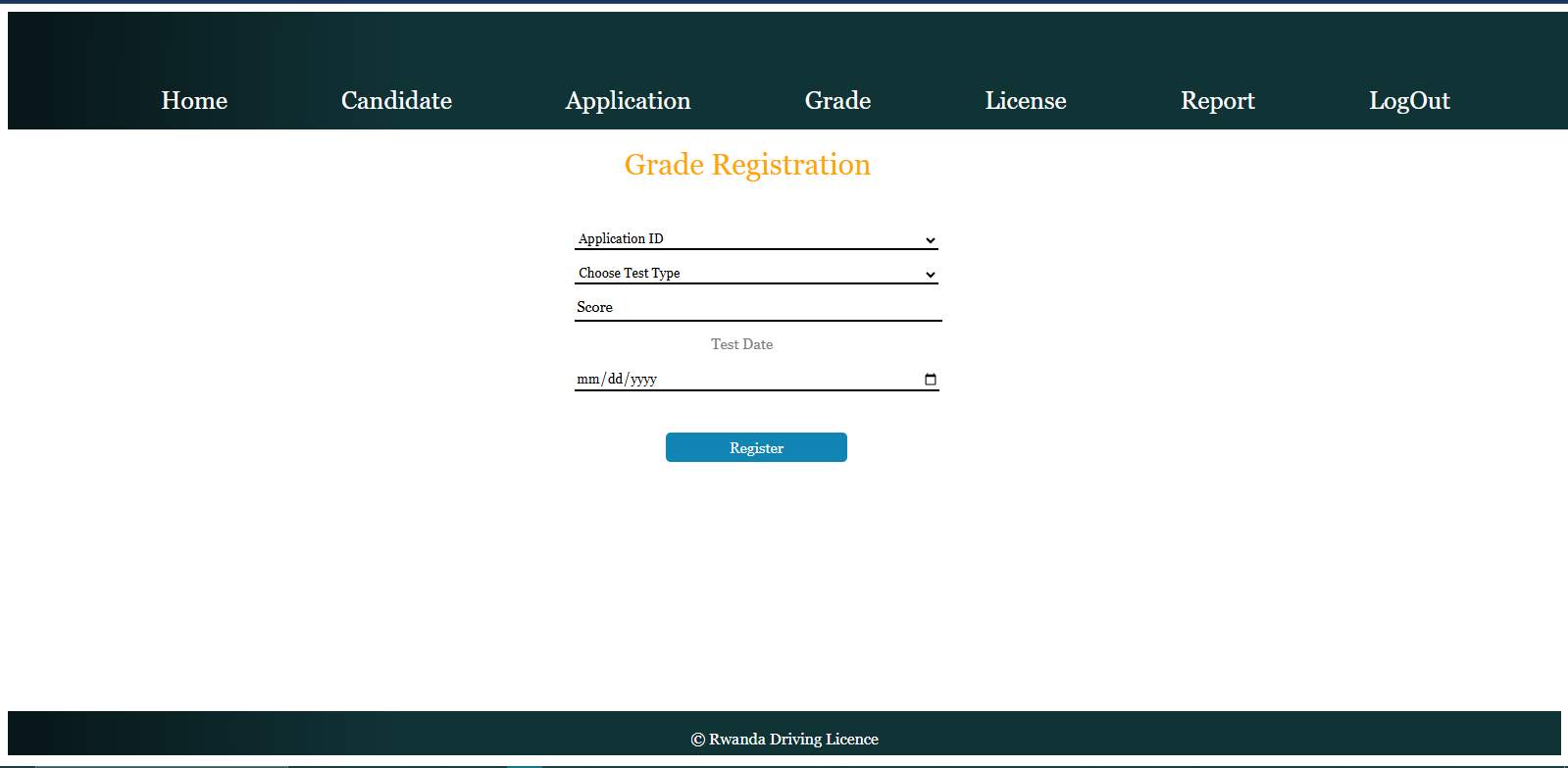
2. Page Load:

* Displays a navigation bar with links to Home, Candidate, Application, Grade, License, Report, and LogOut, including sub-menus for detailed actions.
* Presents a form titled "Grade Registration" with fields: Application ID (drop-down populated from "Application" table), Test Type (Theory/Practical), Score, and Test Date.

3. Form Submission:

* User fills and submits the form by clicking "Register."
* PHP processes:
* Checks the Score: if 60-100, sets Decision to "Pass" and Status to "Approved"; if below 60, sets Decision to "Fail" and Status to "Rejected."
* Inserts the grade data (Application ID, Test Type, Score, Test Date, Decision) into the "Grade" table.
* Updates the "Application" table with the corresponding Status (Approved/Rejected).
* Displays a success ("Registration Successful") or failure ("Registration Failed") alert.

1. User Navigation:Users can navigate to other sections via the navigation bar or log out via "Logout.php."



Grade Registration page provides a functional interface for recording test results and updating application statuses, supported by HTML, CSS, and PHP with MySQL integration. It automates Pass/Fail decisions based on scores and syncs with application records, enhancing process efficiency. However, security risks (e.g., lack of prepared statements for SQL injection prevention, invalidated inputs) and limited validation (e.g., no score range enforcement beyond 60) suggest improvements are needed for security and reliability.

# Grade Overview

Grade Overview page allows authenticated users to view a comprehensive list of all grade records, displaying details such as Application ID, National ID, Test Type, Score, Decision, and Test Date. It provides options to update or delete grade records, facilitating efficient management of test results. This page supports the system's goal of streamlining the licensing process by offering a centralized view for monitoring and maintaining grade data.

## Functional Flow

1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

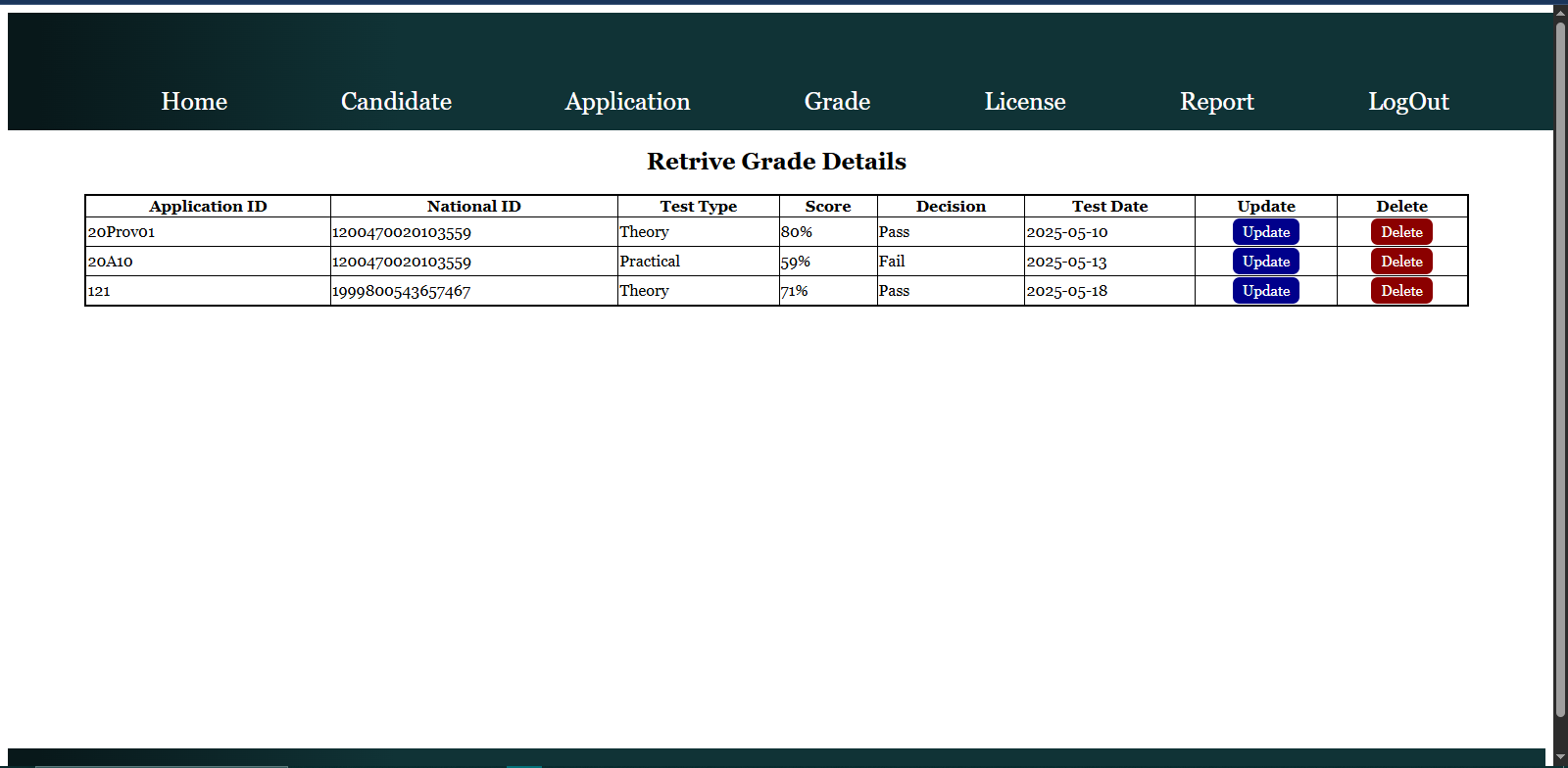
2. Page Load:

* Displays a navigation bar with links to Home, Candidate, Application, Grade, License, Report, and LogOut, including sub-menus for detailed actions.
* Executes a JOIN query on the "Grade" and "Application" tables to retrieve grade details linked with National IDs.
* Presents the data in a table with columns: Application ID, National ID, Test Type, Score (with % symbol), Decision, Test Date, Update, and Delete.

3. Data Display and Actions: Each row represents a grade record, populated with its details.

* Update Action:A button links to "GradeUpdate.php" with the Application ID as a parameter , prompting a confirmation ("Are Sure You Want to Update?").
* Delete Action:A button links to "Gradedelete.php" with the Application ID as a parameter, prompting a confirmation ("Are Sure You Want to Delete?").

1. User Navigation:Users can navigate to other sections via the navigation bar or log out via "Logout.php."



Grade Overview page provides a clear and functional interface for viewing and managing grade records, supported by HTML, CSS, and PHP with MySQL integration. It enhances administrative control with update and delete functionalities, using a JOIN query to include National IDs. However, security risks (e.g., lack of prepared statements for SQL queries, unencrypted GET parameters) and minimal styling suggest improvements are needed for security and user experience.

## Grade Deletion

Grade Deletion functionality allows authenticated users to remove grade records from the system using an Application ID. It ensures data integrity by checking for dependencies in related tables (Application, License, and Grade) before deletion, preventing orphaned records and maintaining database consistency. This feature supports the system's goal of efficiently managing test result data within the licensing process.

## Functional Flow

1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

2. Deletion Request:

* Triggered when a delete request is initiated with an Application ID passed via a GET parameter.
* Sanitizes the Application ID using `MySQL\_real\_escape\_string` to prevent SQL injection.

3. Dependency Check:

* Queries the "Application," "License," and "Grade" tables to check for dependencies:
* "Application" table: Verifies if the Application ID exists.
* "License" table: Checks if the Application ID is linked.
* "Grade" table: Checks for grades associated with the National ID linked to the Application ID.
* If any dependencies are found, displays an alert ("Cannot delete. This ID is in use to other records.") and redirects to "VCandidate.php."

4. Deletion Process:

* If no dependencies exist, deletes the grade record from the "Grade" table using the Application ID.
* Displays a success ("Deleted successfully") or failure ("Delete failed") alert and redirects to "VGrade.php."

5. Error Handling:If the parameter is not set, no action is taken (implicitly handled by the `isset` condition).

Grade Deletion functionality ensures safe removal of grade records by validating dependencies across multiple tables, supported by PHP and MySQL. It prevents data inconsistencies and redirects appropriately post-action. However, security risks (e.g., reliance on `MySQL\_real\_escape\_string` alone for SQL injection prevention, unencrypted session handling) and incomplete error handling (e.g., no explicit invalid request alert) suggest improvements are needed for security and robustness.

## Grade Update

Grade Update page allows authenticated users to modify existing grade records in the system. It retrieves a grade record using its Application ID (passed via a GET parameter) and enables updates to fields like Test Type, Score, and Test Date, automatically determining a Pass/Fail Decision and updating the associated application status (Approved/Rejected). This feature supports the system's goal of maintaining an accurate and efficient licensing process by allowing users to correct or update test results.

## Functional Flow

1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

2. Page Load:

* Displays a navigation bar with links to Home, Candidate, Application, Grade, License, Report, and LogOut, including sub-menus for detailed actions.
* Retrieves the grade record from the "Grade" table using the Application ID and populates the form with existing values.
* Form fields include Application ID (pre-filled, read-only), Test Type (drop-down: Theory/Practical), Score, and Test Date.

1. Form Submission:User updates the fields and clicks "Update."

**PHP processes:**

* Evaluates the Score: if 60-100, sets Decision to "Pass" and Status to "Approved"; if below 60, sets Decision to "Fail" and Status to "Rejected."
* Updates the "Grade" table with the new Test Type, Score, Test Date, and Decision for the specified Application ID.
* Updates the "Application" table with the corresponding Status (Approved/Rejected).
* Displays a success ("Grade Update Successful") or failure ("Grade Update Failed") alert and redirects to "VGrade.php."

4. User Navigation:Users can navigate to other sections via the navigation bar or log out via "Logout.php."

Grade Update page provides a straightforward interface for modifying grade records, supported by HTML, CSS, and PHP with MySQL integration. It ensures consistency by syncing grade decisions with application statuses and redirects users to the overview page post-update. However, security risks (e.g., lack of prepared statements for SQL injection prevention, invalidated GET parameter) and limited validation (e.g., no strict score range enforcement) suggest improvements are needed for security and reliability.

# License Registration

License Registration page allows authenticated users to register new driving licenses for candidates. It collects details such as License ID, National ID, License Number, Issue Date, and Expiry Date, ensuring eligibility by validating passing grades and enforcing data constraints. It also auto-updates license statuses to "Expired" if the Expiry Date has passed, supporting the system's goal of managing the licensing process efficiently and accurately.

## Functional Flow

1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

2. Page Load:

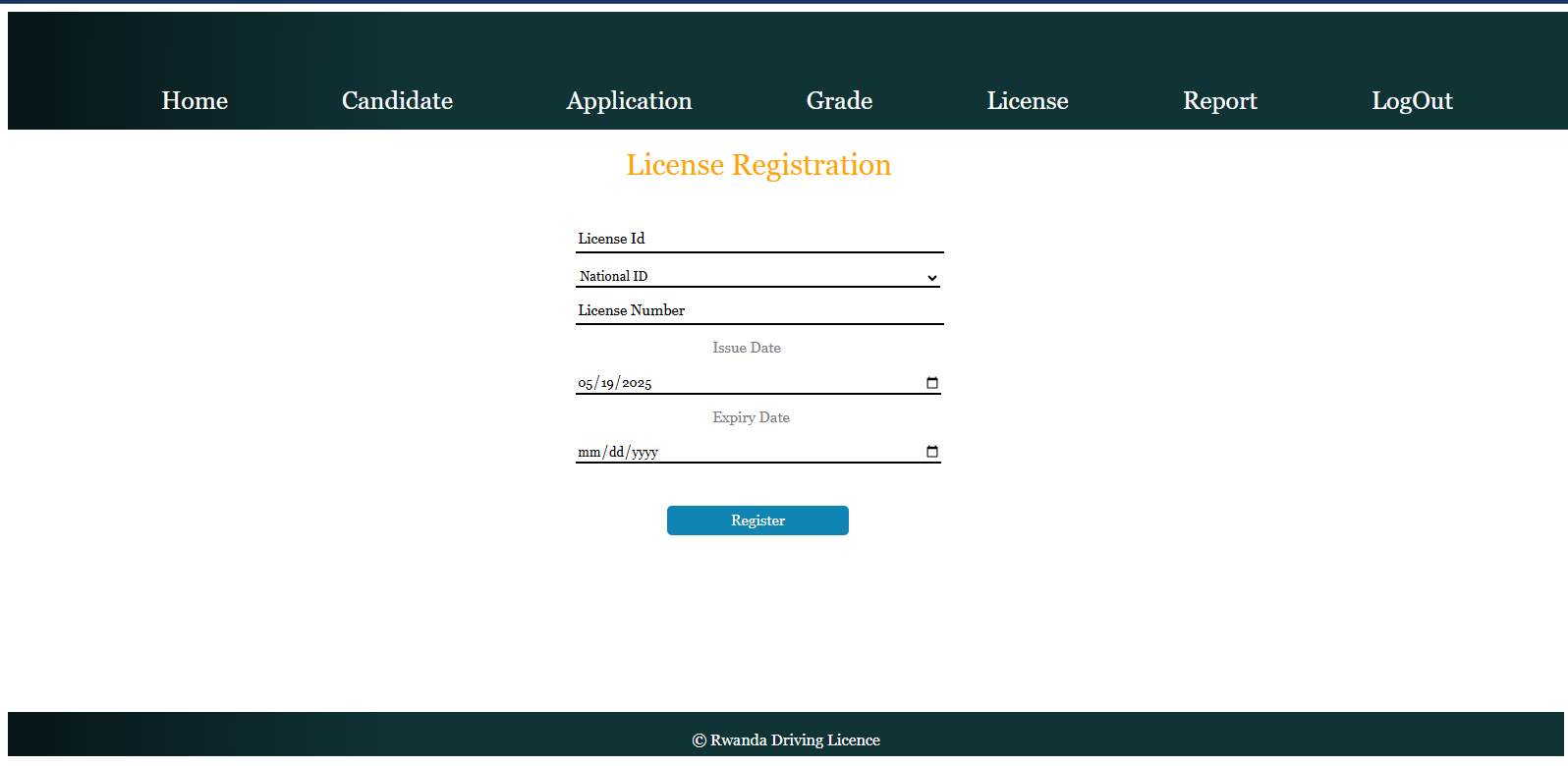
* Displays a navigation bar with links to Home, Candidate, Application, Grade, License, Report, and LogOut, including sub-menus.
* Presents a form titled "License Registration" with fields: License ID, National ID (drop-down populated via a JOIN on "Candidate" and "Application" tables), License Number, Issue Date (pre-filled with today’s date), and Expiry Date.
* Auto-updates the "License" table by setting the Status to "Expired" for records where the Expiry Date is before the current date (`CURDATE()`).

1. Form Submission:User fills and submits the form by clicking "Register.

**PHP processes:**

* Validates License Number length (must be exactly 16 characters).
* Checks for duplicates (License ID, License Number, or National ID) in the "License" table.
* Ensures Expiry Date is after Issue Date.
* Verifies candidate eligibility by checking for passing grades (Score >= 60) in the "Grade" table linked via "Application."
* Sets Status to "Active" or "Expired" based on Expiry Date comparison with the current date.
* Inserts the license data into the "License" table.
* Displays a success ("License registered successfully.") or failure alert (e.g., "License registration failed," "Candidate is not eligible").

1. User Navigation:Users can navigate to other sections via the navigation bar or log out via "Logout.php."



License Registration page provides a robust interface for registering licenses, supported by HTML, CSS, and PHP with MySQL integration. It ensures eligibility and data integrity through multiple validations and auto-updates license statuses, enhancing process reliability. However, security risks (e.g., lack of prepared statements for SQL injection prevention, invalidated inputs) and limited user feedback suggest improvements are needed for security and user experience.

# License Overview

License Overview page allows authenticated users to view a comprehensive list of all registered license records, displaying details such as License ID, National ID, Full Name, License Number, Issue Date, Expiry Date, and Status. It provides options to update or delete license records, facilitating effective management of license data. This page supports the system's goal of streamlining the licensing process by offering a centralized view for monitoring and maintaining license information.

## Functional Flow

1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

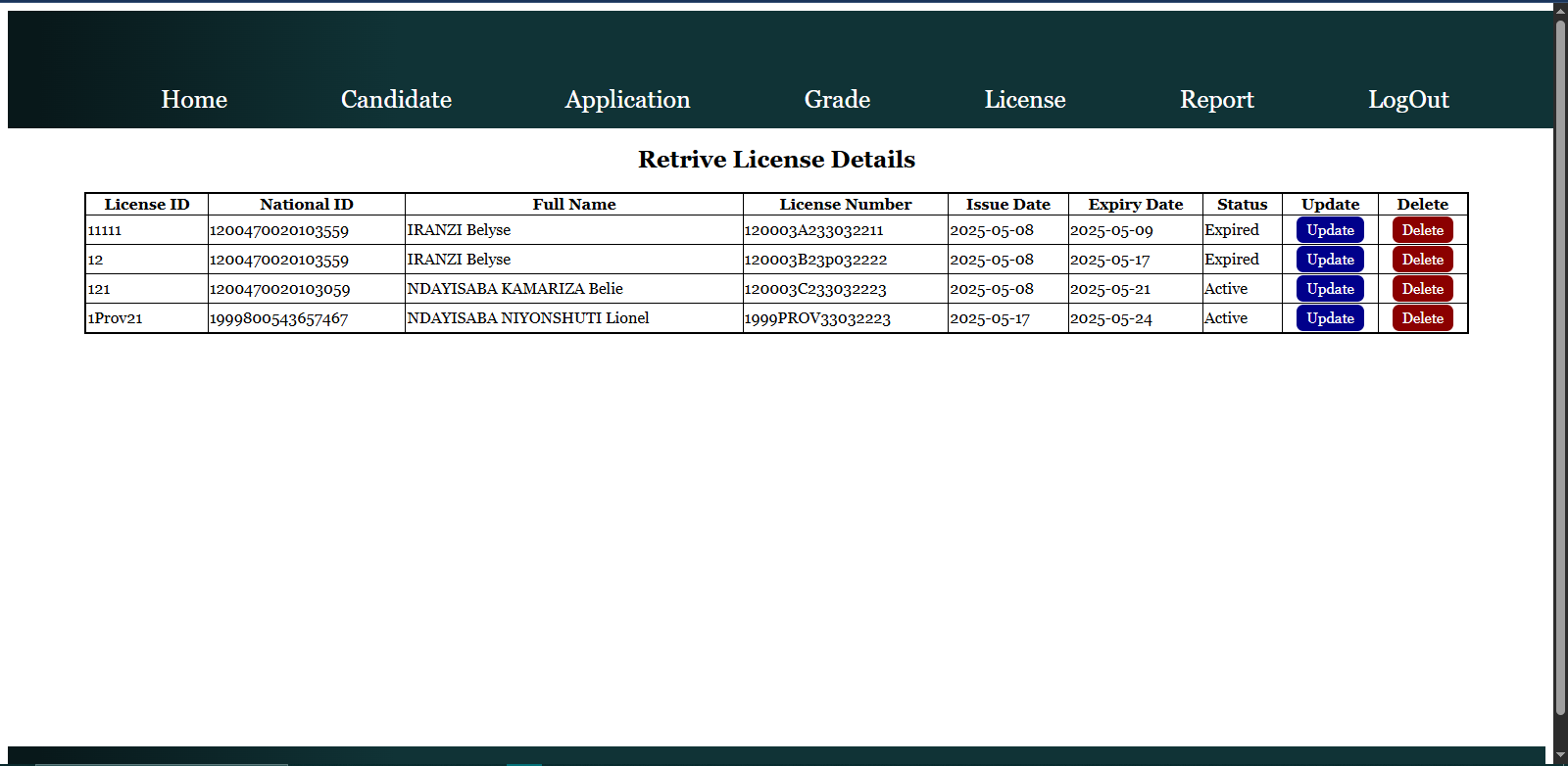
2. Page Load:

* Displays a navigation bar with links to Home, Candidate, Application, Grade, License, Report, and LogOut, including sub-menus for detailed actions.
* Executes a JOIN query on the "License" and "Candidate" tables to retrieve license details linked with candidate names.
* Presents the data in a table with columns: License ID, National ID, Full Name, License Number, Issue Date, Expiry Date, Status, Update, and Delete.

3. Data Display and Actions:Each row represents a license record, populated with its details.

* Update Action: A button links to "LicenseUpdate.php" with the License ID as a parameter, prompting a confirmation ("Are Sure You Want to Update?").
* Delete Action: A button links to "Licensedelete.php" with the License ID as a parameter, prompting a confirmation ("Are Sure You Want to Delete?").

1. User Navigation:Users can navigate to other sections via the navigation bar or log out via "Logout.php."



License Overview page provides a clear and functional interface for viewing and managing license records, supported by HTML, CSS, and PHP with MySQL integration. It enhances administrative control with update and delete functionalities, using a JOIN query to include candidate names. However, security risks (e.g., lack of prepared statements for SQL queries, unencrypted GET parameters) and minimal styling suggest improvements are needed for security and user experience.

## **License Update**

License Update page allows authenticated users to modify existing license records in the system. It retrieves a license record using its License ID (passed via a GET parameter) and enables updates to fields like License Number, Issue Date, and Expiry Date, automatically determining the Status (Active/Expired) based on the Expiry Date. This feature supports the system's goal of maintaining an accurate and efficient licensing process by allowing users to correct or update license details.

## Functional Flow

1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

2. Page Load:

* Displays a navigation bar with links to Home, Candidate, Application, Grade, License, Report, and LogOut, including sub-menus.
* Retrieves the license record from the "License" table using the License ID and populates the form with existing values.
* Form fields include License ID (pre-filled, read-only), National ID, License Number, Issue Date, and Expiry Date.
* Auto-updates the "License" table by setting the Status to "Expired" for records where the Expiry Date is before the current date (`CURDATE()`).

1. Form Submission:User updates the fields and clicks "Update."

**PHP processes:**

* Validates License Number length (must be exactly 16 characters).
* Ensures Expiry Date is after Issue Date.
* Determines Status based on Expiry Date comparison with the current date (Expired if before today, Active otherwise).
* Updates the "License" table with the new License Number, Issue Date, Expiry Date, and Status for the specified License ID.
* Displays a success ("License Update successfully.") or failure ("License Update failed.") alert and redirects to "VLicense.php."

4. User Navigation:Users can navigate to other sections via the navigation bar or log out via "Logout.php."

License Update page provides a functional interface for modifying license records, supported by HTML, CSS, and PHP with MySQL integration. It ensures data consistency with validations and auto-updates statuses, redirecting users to the overview page post-update. However, security risks (e.g., lack of prepared statements for SQL injection prevention, invalidated GET parameter) and limited validation (e.g., no duplicate checks or National ID validation) suggest improvements are needed for security and reliability.

## License Deletion

License Deletion functionality allows authenticated users to remove license records from the system using their License ID. This feature supports the system's goal of maintaining an accurate and manageable database for the licensing process by enabling the removal of license records when necessary.

## Functional Flow

1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

2. Deletion Request:

* Triggered when a delete request is initiated with a License ID passed via a GET parameter (`$\_GET['del']`).
* Sanitizes the License ID using `MySQL\_real\_escape\_string` to prevent SQL injection.

3. Deletion Process:

* Deletes the license record from the "License" table using the provided License ID.
* Displays a success ("Deleted successfully") or failure ("Delete failed") alert and redirects to "VLicense.php" (License Overview page).

4. Error Handling:If the parameter is not set, displays an alert ("Invalid request").

License Deletion functionality provides a straightforward method for removing license records, supported by PHP and MySQL. It ensures users can delete records with minimal steps and redirects them to the overview page post-action. However, it lacks dependency checks (e.g., related records in other tables), which could lead to data inconsistencies. Security risks (e.g., reliance on `MySQL\_real\_escape\_string` alone for SQL injection prevention, unencrypted session handling) also suggest improvements are needed for security and data integrity.

# Candidate Report

Candidate Report page allows authenticated users to select a candidate by their National ID and generate a report. It provides a drop-down of all candidates with their National IDs and full names, enabling users to initiate a reporting process for a specific candidate. This feature supports the system's goal of providing detailed insights into candidate data, facilitating administrative oversight and reporting within the licensing process.

## Functional Flow

1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

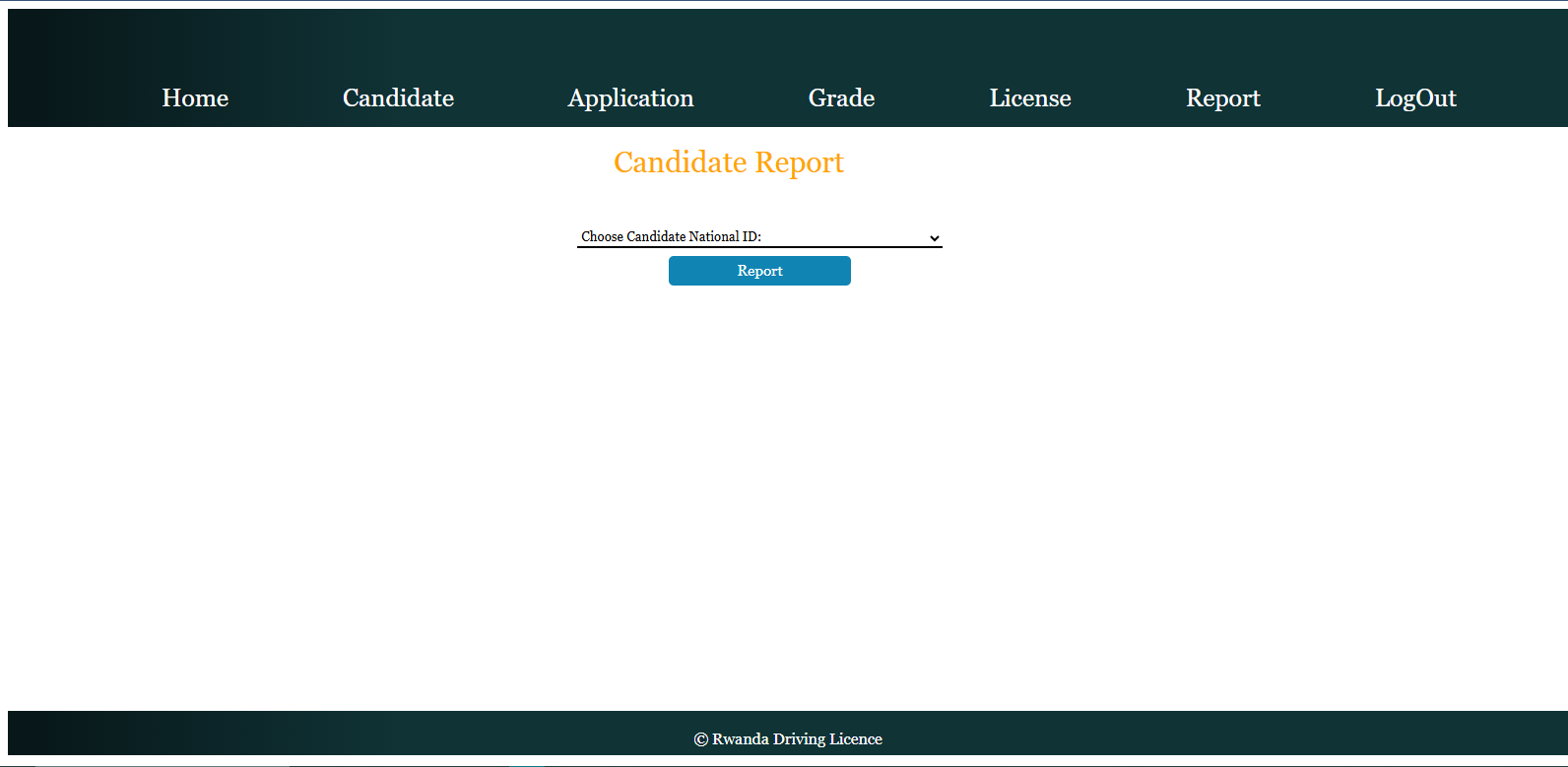
2. Page Load:

* Displays a navigation bar with links to Home, Candidate, Application, Grade, License, Report, and LogOut, including sub-menus for detailed actions.
* Presents a form titled "Candidate Report" with a drop-down menu populated by querying the "Candidate" table, showing each candidate’s National ID and Full Name.

3. Form Submission:

* User selects a National ID from the drop-down and clicks "Report."
* Submits the form via POST to "Vsingle.php," passing the selected National ID for further processing (e.g., generating a detailed candidate report).

1. User Navigation:Users can navigate to other sections via the navigation bar or log out via "Logout.php."



Candidate Report page provides a simple interface for initiating candidate-specific reports, supported by HTML, CSS, and PHP with MySQL integration. It enables users to select a candidate and proceed to a detailed report view, enhancing administrative reporting capabilities. However, security risks (e.g., lack of prepared statements for SQL queries, no input validation) and minimal error handling (e.g., no feedback if no candidate is selected) suggest improvements are needed for security and user experience.

# Candidate Report Display

Candidate Report Display page allows authenticated users to view a detailed report for a specific candidate, identified by their National ID. It displays candidate and application details, including National ID, Application ID, License Type, Full Name, Gender, Birth Date, Email, and Phone Number, in a tabular format. This feature supports the system's goal of providing comprehensive reporting capabilities for administrative oversight within the licensing process.

## Functional Flow

1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

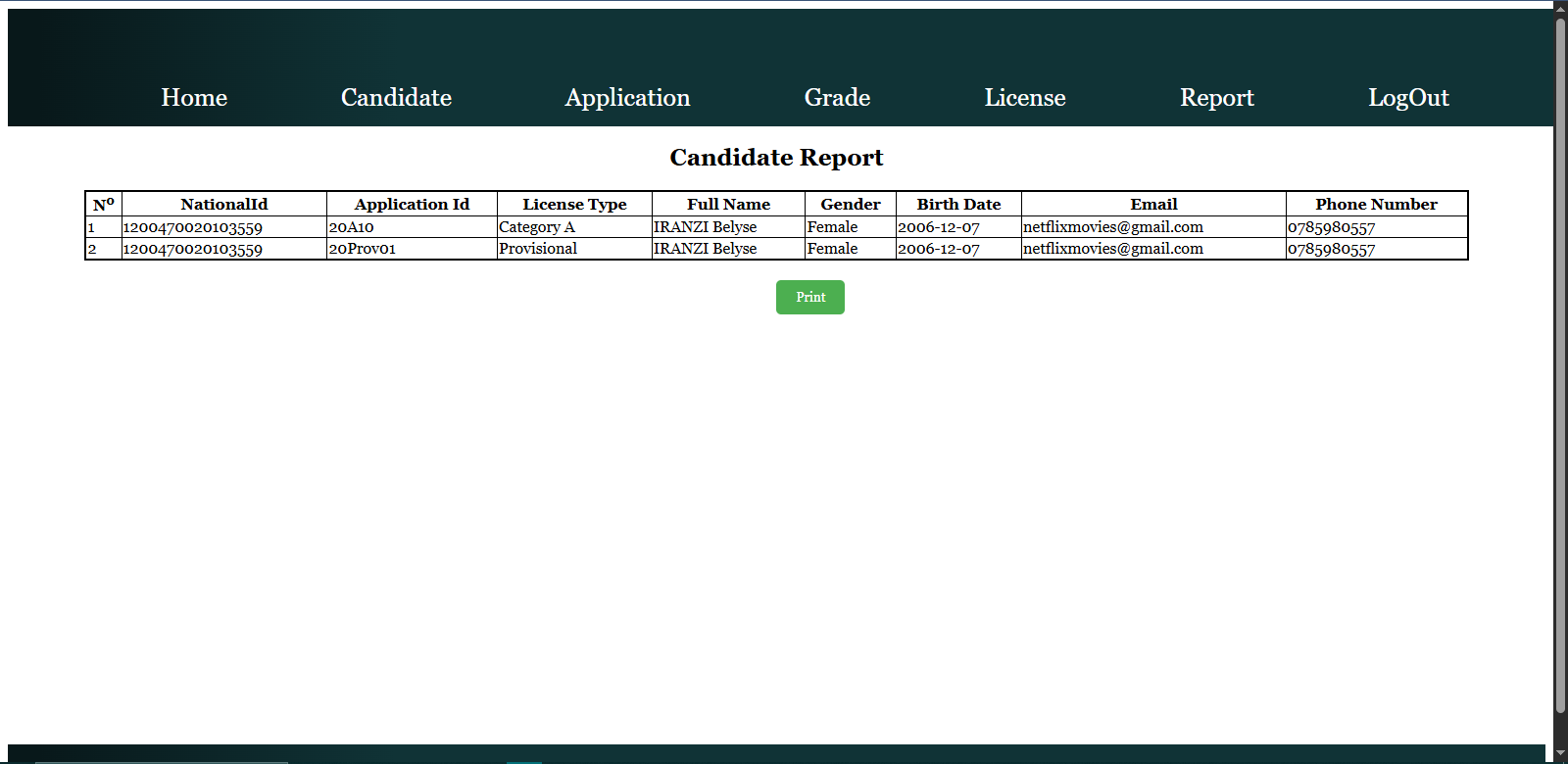
2. Page Load:

* Displays a navigation bar with links to Home, Candidate, Application, Grade, License, Report, and LogOut, including sub-menus for detailed actions.
* Checks for form submission (`$\_POST['ok']`) containing the selected National ID (`$\_POST['NationalId']`) from the Candidate Report page (Single.php).

3. Data Retrieval and Display:

* Executes a JOIN query on the "Candidate" and "Application" tables to retrieve records matching the provided National ID.
* Presents the data in a table with columns: N<sup>o</sup> (row counter), National ID, Application ID, License Type, Full Name, Gender, Birth Date, Email, and Phone Number.
* If no records are found, displays a message ("No data available") spanning all columns.
* If records exist, iterates through the results, incrementing a counter (`$c`) for each row and populating the table.

1. User Navigation:Users can navigate to other sections via the navigation bar or log out via "Logout.php."



Candidate Report Display page provides a clear interface for viewing candidate-specific reports, supported by HTML, CSS, and PHP with MySQL integration. It effectively consolidates candidate and application data using a JOIN query, enhancing administrative reporting capabilities. However, security risks (e.g., lack of prepared statements for SQL queries, invalidated POST data) and minimal error handling (e.g., no redirect or message if the form isn’t submitted) suggest improvements are needed for security and user experience.

# Application Report

Application Report page allows authenticated users to generate a report of application records within a specified date range. Users can input a Start Date and End Date to filter the data, enabling detailed insights into application activities. This feature supports the system's goal of providing comprehensive reporting capabilities for administrative oversight within the licensing process.

## Functional Flow

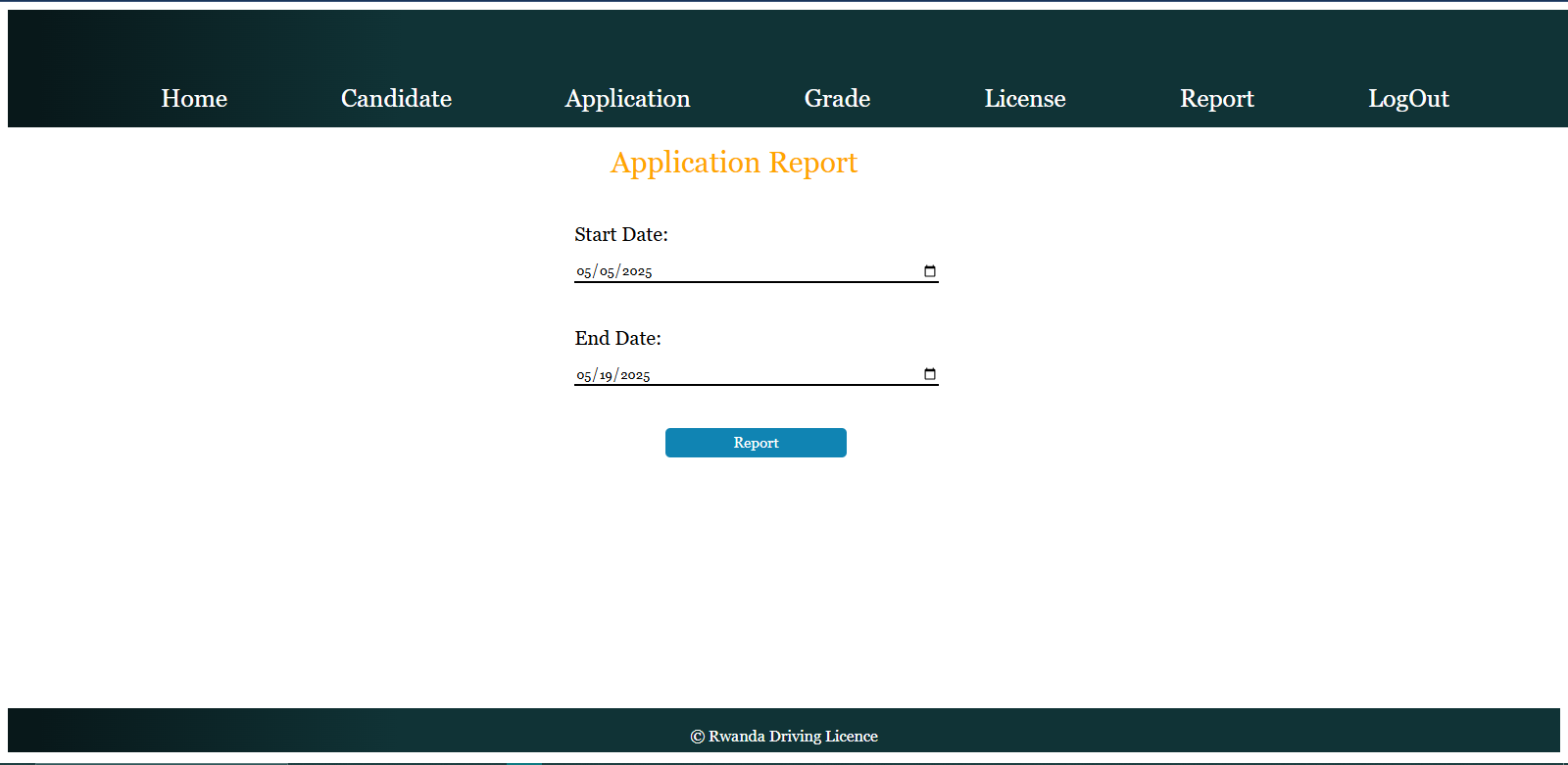
1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.
2. Page Load:

* Displays a navigation bar with links to Home, Candidate, Application, Grade, License, Report, and LogOut, including sub-menus for detailed actions.
* Presents a form titled "Application Report" with fields for Start Date and End Date (date input fields).

1. Form Submission:

* User enters a Start Date and End Date and clicks "Report."
* Submits the form via POST to "VReportApp.php," passing the selected dates (`date1` and `date2`) for further processing.

1. User Navigation:Users can navigate to other sections via the navigation bar or log out via "Logout.php."



Application Report page provides a simple interface for initiating date-filtered application reports, supported by HTML, CSS, and PHP with MySQL integration. It enables users to define a date range for reporting, enhancing administrative reporting capabilities. However, security risks (e.g., lack of input validation for dates, no prepared statements for SQL queries) and minimal error handling (e.g., no feedback if dates are invalid or missing) suggest improvements are needed for security and user experience.

# Application Report Display

Application Report Display page allows authenticated users to view a filtered report of application records within a specified date range. It displays details such as National ID, Application ID, Application Date, License Type, and Status in a tabular format, providing insights into application activities. This feature supports the system's goal of offering detailed reporting capabilities for administrative oversight.

## Functional Flow

1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

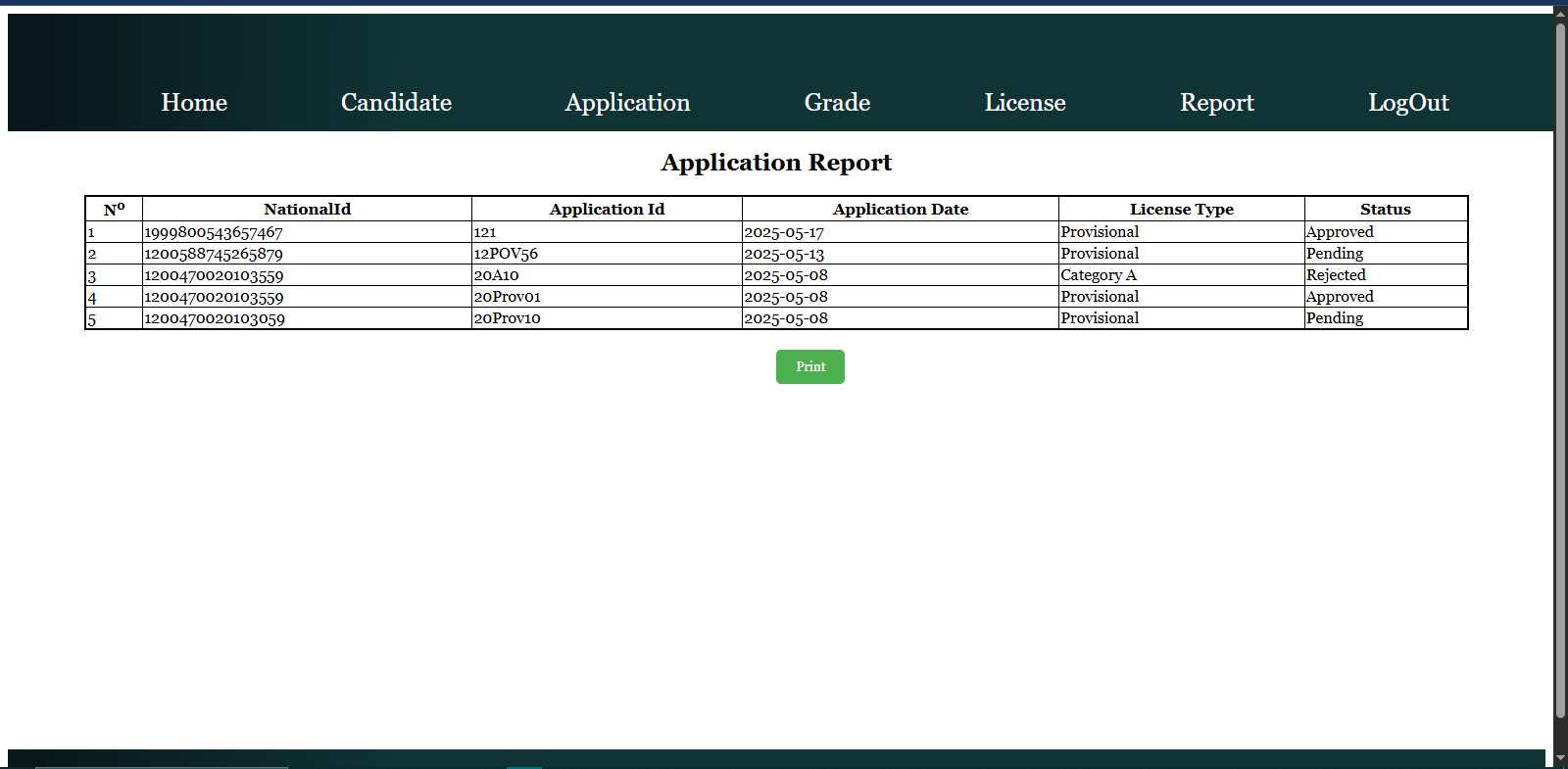
2. Page Load:

* Displays a navigation bar with links to Home, Candidate, Application, Grade, License, Report, and LogOut, including sub-menus.
* Checks for form submission (`$\_POST['ok']`) containing the selected Start Date (`$\_POST['date1']`) and End Date (`$\_POST['date2']`) from the Application Report page.

1. Data Retrieval and Display: Establishes a connection to the MySQL database.

* Executes a query on the "Application" table to retrieve records where the Application Date falls between the specified Start Date and End Date using the `BETWEEN` clause.
* Presents the data in a table with columns: N<sup>o</sup> (row counter), National ID, Application ID, Application Date, License Type, and Status.
* If no records are found, displays a message ("No data available") spanning all columns.
* If records exist, iterates through the results, incrementing a counter (`$c`) for each row and populating the table.

1. User Navigation: Users can navigate to other sections via the navigation bar or log out via "Logout.php."



Application Report Display page provides a clear interface for viewing date-filtered application reports, supported by HTML, CSS, and PHP with MySQL integration. It effectively filters and presents application data based on date ranges, enhancing administrative reporting capabilities. However, security risks (e.g., lack of prepared statements for SQL injection, invalidated POST data) and minimal error handling (e.g., no validation for date range consistency or database connection errors) suggest improvements are needed for security and reliability.

# Grade Report

The Rwanda Driving License Grade Report page allows authenticated users to generate a report of grade records filtered by a specific decision (e.g., Pass or Fail). It provides a drop-down menu populated with unique decision values from the "Grade" table, enabling users to select a decision and initiate a detailed report. This feature supports the system's goal of providing targeted reporting capabilities for administrative oversight within the licensing process.

## Functional Flow

1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

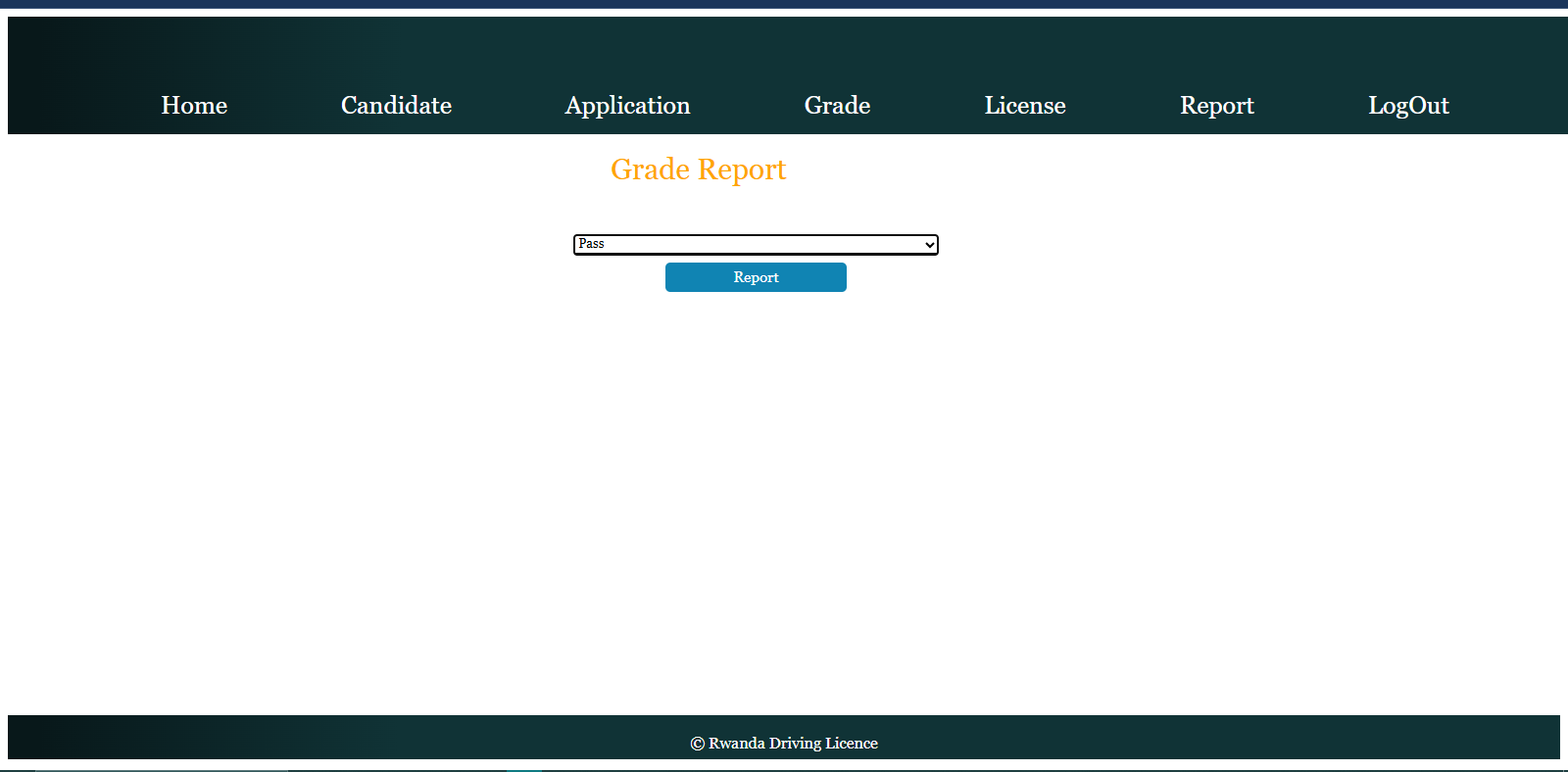
2. Page Load:

* Displays a navigation bar with links to Home, Candidate, Application, Grade, License, Report, and LogOut, including sub-menus for detailed actions.
* Presents a form titled "Grade Report" with a drop-down menu populated by querying distinct "Decision" values from the "Grade" table.

3. Form Submission:

* User selects a decision from the drop-down and clicks "Report."
* Submits the form via POST to "VDecision.php," passing the selected decision (`decision`) for further processing (e.g., generating a filtered grade report).

1. User Navigation:Users can navigate to other sections via the navigation bar or log out via "Logout.php."



Grade Report page provides a simple interface for initiating decision-filtered grade reports, supported by HTML, CSS, and PHP with MySQL integration. It leverages a distinct query to populate the decision options, enhancing administrative reporting capabilities. However, security risks and minimal error handling (e.g., no feedback if no decision is selected) suggest improvements are needed for security and user experience.

# Grade Decision Report Display

Grade Decision Report Display page allows authenticated users to view a filtered report of grade records based on a selected decision (e.g., Pass or Fail). It displays details such as Application ID, Test Type, Score, Decision, and Test Date in a tabular format, providing insights into test outcomes. This feature supports the system's goal of offering targeted reporting capabilities for administrative oversight within the licensing process.

## Functional Flow

1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

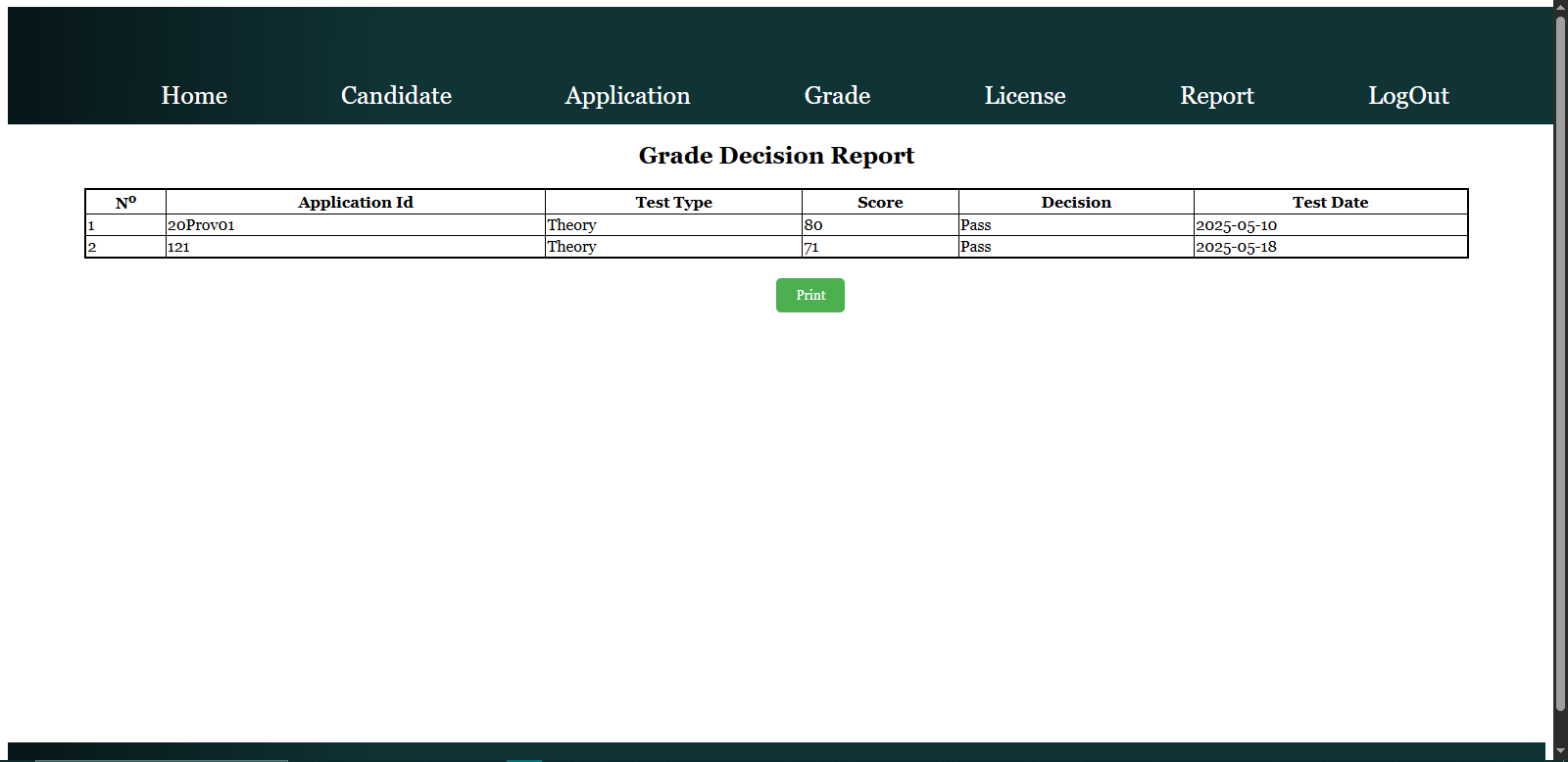
2. Page Load:

* Displays a navigation bar with links to Home, Candidate, Application, Grade, License, Report, and LogOut, including sub-menus for detailed actions.
* Checks for form submission (`$\_POST['ok']`) containing the selected decision (`$\_POST['decision']`) from the Grade Report page (Decision.php).

3. Data Retrieval and Display:

* Executes a query on the "Grade" table to retrieve records matching the selected Decision.
* Presents the data in a table with columns: N<sup>o</sup> (row counter), Application ID, Test Type, Score, Decision, and Test Date.
* If no records are found, displays a message ("No data available") spanning all columns.
* If records exist, iterates through the results, incrementing a counter (`$c`) for each row and populating the table.

1. User Navigation:Users can navigate to other sections via the navigation bar or log out via "Logout.php."



Grade Decision Report Display page provides a clear interface for viewing decision-filtered grade reports, supported by HTML, CSS, and PHP with MySQL integration. It effectively filters and presents grade data, enhancing administrative reporting capabilities. However, security risks (e.g., lack of prepared statements for SQL queries, invalidated POST data) and minimal error handling (e.g., no redirect or message if the form isn’t submitted) suggest improvements are needed for security and user experience.

# License Report

License Report page allows authenticated users to generate a report of license records filtered by a specific status (e.g., Active or Expired). It provides a drop-down menu populated with unique status values from the "License" table, enabling users to select a status and initiate a detailed report. This feature supports the system's goal of providing targeted reporting capabilities for administrative oversight within the licensing process.

## Functional Flow

1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

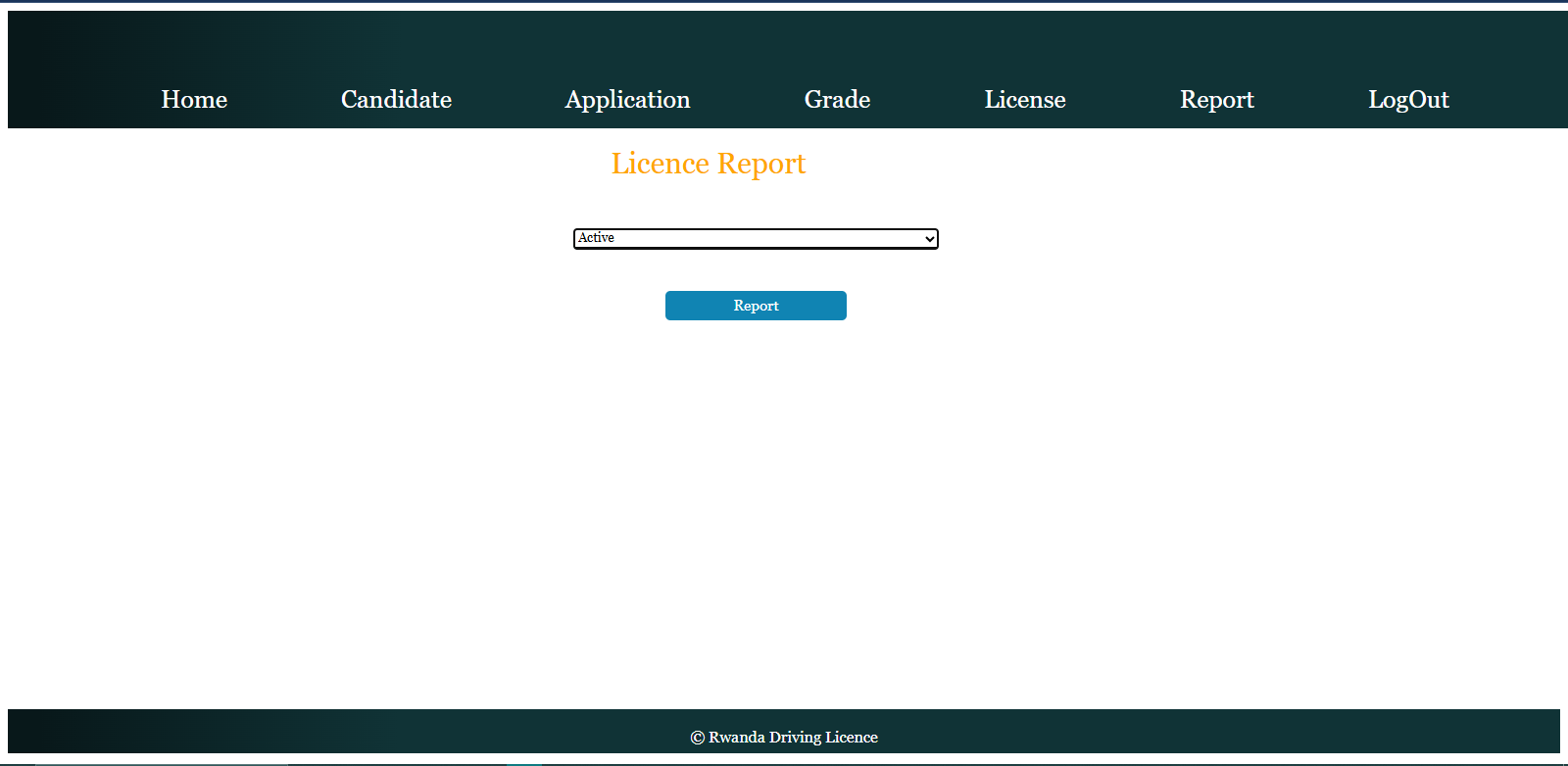
2. Page Load:

* Displays a navigation bar with links to Home, Candidate, Application, Grade, License, Report, and LogOut, including sub-menus for detailed actions.
* Presents a form titled "Licence Report" with a drop-down menu populated by querying distinct "Status" values from the "License" table.

3. Form Submission:

* User selects a status from the drop-down and clicks "Report."
* Submits the form via POST to "VreportLicense.php," passing the selected status (`status`) for further processing (e.g., generating a filtered license report).

1. User Navigation:Users can navigate to other sections via the navigation bar or log out via "Logout.php."



License Report page provides a simple interface for initiating status-filtered license reports, supported by HTML, CSS, and PHP with MySQL integration. It leverages a distinct query to populate the status options, enhancing administrative reporting capabilities. However, security risks (e.g., lack of prepared statements for SQL queries, no input validation) and minimal error handling (e.g., no feedback if no status is selected) suggest improvements are needed for security and user experience.

# License Report Display

License Report Display page allows authenticated users to view a filtered report of license records based on a selected status (e.g., Active or Expired). It displays details such as License ID, National ID, License Number, Issue Date, Expiry Date, and Status in a tabular format, providing insights into license statuses. This feature supports the system's goal of offering targeted reporting capabilities for administrative oversight within the licensing process.

## Functional Flow

1. User Authentication:Verifies an active session (`$\_SESSION['login']`); redirects to "login.php" if absent.

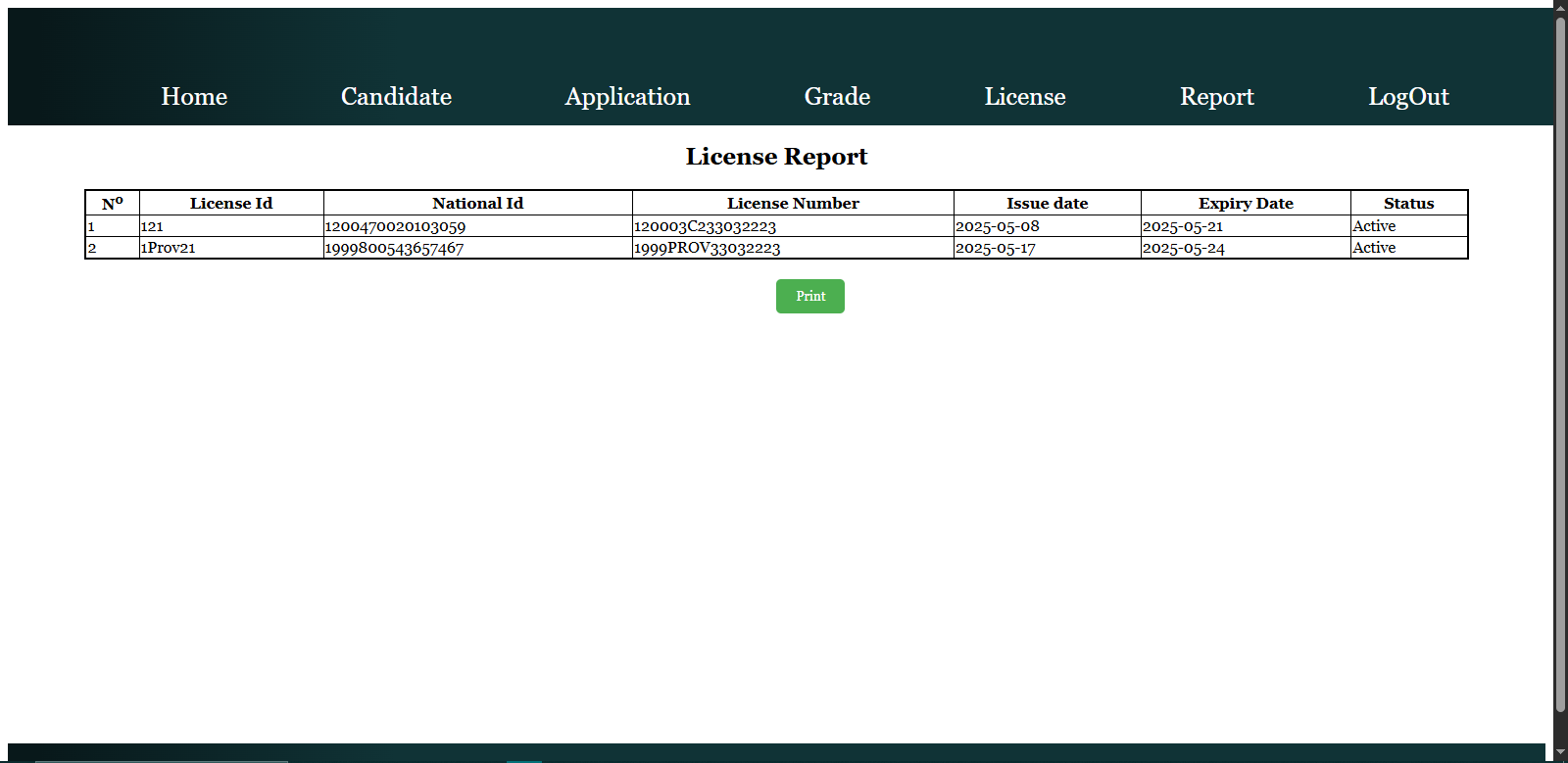
2. Page Load:

* Displays a navigation bar with links to Home, Candidate, Application, Grade, License, Report, and LogOut, including sub-menus for detailed actions.
* Checks for form submission (`$\_POST['ok']`) containing the selected status (`$\_POST['status']`) from the License Report page.

3. Data Retrieval and Display:

* Executes a query on the "License" table to retrieve records matching the selected Status.
* Presents the data in a table with columns: N<sup>o</sup> (row counter), License ID, National ID, License Number, Issue Date, Expiry Date, and Status.
* If no records are found, displays a message ("No data available") spanning all columns.
* If records exist, iterates through the results, incrementing a counter (`$c`) for each row and populating the table.

1. User Navigation:Users can navigate to other sections via the navigation bar or log out via "Logout.php."



License Report Display page provides a clear interface for viewing status-filtered license reports, supported by HTML, CSS, and PHP with MySQL integration. It effectively filters and presents license data, enhancing administrative reporting capabilities. However, security risks (e.g., lack of prepared statements for SQL queries, invalidated POST data) and minimal error handling (e.g., no redirect or message if the form isn’t submitted) suggest improvements are needed for security and user experience.

# Conclusion

The **Rwanda Driving License Management System** effectively transforms a traditionally manual and paper-based process into a modern, web-based solution. By automating candidate registration, application processing, test grading, and license issuance, the system enhances operational efficiency and service delivery.

This project demonstrates how digital solutions can reduce administrative workload, minimize human error, and improve transparency. Through secure data handling, user-friendly interfaces, and automated workflows, the system ensures a reliable and scalable platform for managing driving licenses in Rwanda.

In summary, this system not only improves the experience for citizens but also empowers administrative bodies to work more efficiently and maintain accurate, up-to-date license records.