

# **Developer Documentation: Ministry of Natural Resources and Forestry Canada MFTIP and CLTIP Database Management System**

**COIS 4000Y**

**GROUP - 2**

**Members of the Group**

**Devashish Kapoor (Student ID: 0727519)**

**Amber Ahmed (Student ID: 0680481)**

**Liang Chu (Student ID: 0719153)**

## Table of Contents

<b>Overview</b>	<b>3</b>
<b>Target Audience</b>	<b>3</b>
<b>Key Features</b>	<b>3</b>
<b>Technology Stack</b>	<b>3</b>
1. Frontend:	3
2. Backend:	3
3. APIs and Libraries:	4
4. Version Control:	4
5. Deployment and Hosting:	4
<b>Directory Structure</b>	<b>4</b>
Home Page:	4
Landowner Landing Page (landowner-landing.php):	6
MFTIP Home:	8
Managed Forest Tax Incentive Program (MFTIP) and Conservation Land Tax Incentive (CLTIP) Program Requests for Reconsideration Search pages:	9
Managed Forest Plan Approvers exam registration webpage:	10
Exporting Search Results as CSV:	12
CLTIP Home:	13
Connect with Managed Forest Plan Approvers Page:	14
Conservation Land Tax Incentive Program (CLTIP) Records Search (cltip-plan, cltip-properties, cltip-landowners):	16
Managed Forest Tax Incentive Program (MFTIP) Records Search (mftip-plan, mftip-properties, mftip-landowners):	17
MNRF CLTIP & MFTIP Employee Login:	19
Landowner Login Web Page:	20
MFPA Login Page:	21
Logout Web Page:	23
Annual eligible areas Conservation Land Tax Incentive Program:	24
Forgot Password Webpage:	27
Find a Managed Forest Plan Approver Webpage:	29
Find a Landowner in Managed Forest Plan Webpage:	30
Lists to Send to MPAC Webpage:	31
Managed Forest Plan:	33
<b>Core Functionalities:</b>	<b>34</b>
<b>Database Schema:</b>	<b>36</b>
<b>Sequence Diagrams:</b>	<b>41</b>
<b>Security Considerations</b>	<b>44</b>
<b>Deployment and Hosting</b>	<b>44</b>

<b>Testing and Quality Assurance:</b>	<b>44</b>
<b>Future Enhancements and Roadmap</b>	<b>44</b>

## Overview

This website serves as a database management system for the Ministry of Natural Resources and Forestry Canada's Managed Forest Tax Incentive Program (MFTIP) and Conservation Land Tax Incentive Program (CLTIP). The website enables landowners to log in and check the status of their applications, as well as allowing Managed Forest Plan approvers exam candidates to register for their exams and access study materials.

## Target Audience

- Ministry of Natural Resources and Forestry Canada staff
- Landowners participating in the MFTIP and CLTIP programs
- Managed Forest Plan approvers exam candidates

## Key Features

1. Secure user authentication for landowners and exam candidates
2. Access to application status and updates for landowners
3. Exam registration and study materials access for exam candidates
4. Database management tools for ministry staff

## Technology Stack

### 1. Frontend:

- HTML5: Markup language used for structuring and presenting content on the web
- CSS3: Stylesheet language used for describing the look and formatting of a document written in HTML
- JavaScript: Programming language used for web development, allowing interactive features and dynamic content
- jQuery: JavaScript library designed to simplify HTML document traversal, manipulation, event handling, and animation
- Bootstrap: CSS framework for developing responsive and mobile-first websites

### 2. Backend:

- PHP: Server-side scripting language used for web development, can be embedded into HTML
- MySQL: Open-source relational database management system, used to store, manage and retrieve data

### 3. APIs and Libraries:

- JSON: Lightweight data-interchange format that is easy to read and write, often used for data transfer between a server and a web application

### 4. Version Control:

- Git: Distributed version control system used to track changes in source code during software development
- GitHub: Web-based hosting service for Git repositories, providing collaboration features and access control
- Github Link: [https://github.com/yencoder/MNR\\_Database](https://github.com/yencoder/MNR_Database)

### 5. Deployment and Hosting:

- Apache or Nginx: Web servers to host the website and serve content to users
- Let's Encrypt: Free, automated, and open certificate authority to enable HTTPS for the website

## Directory Structure

The website's main directory includes the following subdirectories and files:

- includes: Contains shared PHP files, such as database connection settings, metadata and headers.
- styles: Contains CSS files for styling the website.
- individual pages are listed separately below:

### Home Page:

This web page is for the Ministry of Natural Resources and Forestry. It provides information about the Managed Forest Tax Incentive Program (MFTIP) and the Conservation Land Tax Incentive Program (CLTIP). It also includes login options for public users, employees, Managed Forest Plan Approvers (MFPA), and MFPA exam registration.

### DOCTYPE Declaration

<!DOCTYPE html>

This declaration defines the document type and version of HTML being used, which is HTML5.

### HTML Structure:

```
<html lang="en">
```

The opening **<html>** tag has a **lang** attribute set to "en", indicating that the content is in English.

### Head Section

The head section contains metadata and the font library. It includes:

1. Font Awesome library for icons:

```
<script src="https://kit.fontawesome.com/c51f44d97a.js" crossorigin="anonymous"></script>
```

2. PHP variable for the page title:

```
<?php $page_title = "Ministry of Natural Resources and Forestry"; ?>
```

3. PHP include for metadata:

```
<?php include "../includes/metadata.php" ?>
```

### Body Section

The body section contains the header, content, and footer. It includes:

1. PHP include for the header:

```
<?php include '../includes/header.php'; ?>
```

2. A **div** element with class "content" wrapping the main content of the page:

```
<div class="content">
```

3. An **h2** element with the text "About Us".

4. A **section** element with class "aboutUs" containing two **div** elements with class "mftipInfo" and "cltipInfo", providing information about the MFTIP and CLTIP programs, respectively.
5. PHP code to check if the user is logged in and display login options for public users, employees, MFPA, and MFPA exam registration:

```
<?php // session check for user information if (isset($_SESSION['rollNumber'])) {?> ... <?php }?>
```

6. PHP include for the footer:

```
<?php include "../includes/footer.php" ?>
```

## Landowner Landing Page (landowner-landing.php):

This page serves as the landing page for landowners. It displays their property and plan information and provides options to download planning templates and connect with Managed Forest Plan Approvers.

Overview This is a PHP web page that serves as a landing page for authenticated landowners. It displays the landowner's property information, plan details, and provides options to download a planning template form (PDF) and connect with Managed Forest Plan Approvers.

1. Database Tables

The web page interacts with the following database table:

- **landowners**

The structure of the **landowners** table:

Field	Type	Description
<b>ARN</b>	INT	Auto-increment, primary key for the landowner
<b>first_name</b>	VARCHAR	First name of the landowner

<b>property_info</b>	TEXT	Information about the landowner's property
<b>plan_number</b>	VARCHAR	Plan number assigned to the landowner
<b>approved_by</b>	VARCHAR	Person/Entity who approved the plan
<b>plan_start_date</b>	DATE	Start date of the plan
<b>plan_end_date</b>	DATE	End date of the plan
<b>progress_report_due</b>	DATE	Date when the progress report is due
<b>renewal_plan_due</b>	DATE	Date when the renewal plan is due
<b>status_for_tax_year</b>	VARCHAR	Status of the landowner for the 2020 tax year

## File Structure

The web page includes the following external files:

- './includes/library.php': Contains the connectDB function for connecting to the database.
- './includes/metadata.php': Contains common metadata for HTML head section.
- './includes/header-landowner.php': Contains the header section for the landowner.
- './includes/footer.php': Contains the footer section of the web page.
- 'styles/master.css': Contains the CSS styles for the web page.

## 4. Functions

The web page uses the following PHP functions:

- **session\_start()**: Starts or resumes a session.
- **header()**: Sends a raw HTTP header.
- **connectDB()**: Connects to the database.
- **prepare()**: Prepares a SQL statement for execution.
- **execute()**: Executes the prepared statement.
- **fetch()**: Fetches a row from a result set associated with a PDOStatement object.
- **htmlspecialchars()**: Converts special characters to HTML entities.
- **nl2br()**: Inserts HTML line breaks before all newlines in a string.

5. Workflow

- The web page starts a session and checks if the user is authenticated. If not, it redirects the user to the landowner-login.php page.
- It includes the library.php file and connects to the database using the **connectDB()** function.
- The web page queries the **landowners** table to retrieve the landowner's information using the ARN stored in the session.
- The web page displays the landowner's information, plan details, and provides options to download a planning template form and connect with Managed Forest Plan Approvers.

## MFTIP Home:

Overview: This web page allows users to search for properties, landowners, or MFTIP applications using various inputs like roll number, name, or PIN. The search results are displayed in a table. Users can also navigate to different sections like MPAC list, Requests for Reconsideration, Annual eligible area updates, Properties, MFTIP applications, and Landowners.

Database Tables:

1. Parcel
2. Cltip\_application

Fields in the 'Parcel' table:

- ARN
- Current\_owner

Fields in the 'Cltip\_application' table:

- PIN

PHP Code:

- Sanitizes and trims the user inputs.
- Connects to the database.
- Handles different search cases based on the inputs provided (roll number, landlord name, business name, or PIN).



- Queries the appropriate table with the given input and returns the results.
- If no input is provided, a default view from the 'Parcel' table is displayed.

HTML Structure:

1. Head: Includes page title and metadata.
2. Body: a. Header: Common header for the website. b. Search Section: Contains forms for searching by roll number, landlord name, business name, or PIN. c. Search Results Section: Displays the search results in a table. d. Work Process Section: Provides navigation to MPAC list, Requests for Reconsideration, and Annual eligible area updates. e. Records Section: Provides navigation to Properties, MFTIP applications, and Landowners. f. Footer: Common footer for the website.

## Managed Forest Tax Incentive Program (MFTIP) and Conservation Land Tax Incentive (CLTIP) Program Requests for Reconsideration Search pages:

Description: These webpages allow users to search for MFTIP or CLTIP applications by Assessment Roll Number (ARN) or by landowner name. It displays search results in two tables: Pending Requests for Reconsideration and Completed Requests for Reconsideration. Users can also export the search results as CSV files.

Database Tables:

1. Appeal\_parcel

Columns:

- program (VARCHAR): The program associated with the application (in this case, 'MFTIP-Managed Forest Tax Incentive Program' or 'CLTIP-Conservation Land Tax Incentive Program').
- ARN (VARCHAR): The 15-digit Assessment Roll Number.
- Owner (VARCHAR): The name of the landowner.
- Status (VARCHAR): The status of the application ('Active' or 'Completed').
- Other columns, depending on the schema of the **Appeal\_parcel** table. These columns are fetched dynamically and displayed in the search results.

Flow of the Web Page:

1. Users input an ARN or a landowner name and submit the search form.
2. The PHP script connects to the database and executes a query based on the input.

3. The script fetches search results from the **Appeal\_parcel**s table, filtering by the given ARN or landowner name, the program ('MFTIP-Managed Forest Tax Incentive Program' or 'Conservation Land Tax Incentive Program '), and the status ('Active' or 'Completed').
4. If there are any matching records, the script creates two tables to display the search results: Pending Requests for Reconsideration (Status = 'Active') and Completed Requests for Reconsideration (Status = 'Completed').
5. Users can click buttons to export the search results as CSV files.

Files and Includes:

1. **../includes/library.php**: A file containing helper functions for connecting to the database.
2. **../includes/metadata.php**: A file containing metadata for the web page, such as charset, viewport, and title.
3. **../includes/header.php**: A file containing the header section of the web page.
4. **../includes/footer.php**: A file containing the footer section of the web page.
5. **styles/master.css**: A CSS file containing the styling for the web page.
6. **exportRough.php**: A file that handles exporting search results as CSV files.

Dependencies:

1. PHP version 7.0 or later.
2. A web server with PHP support.
3. A database server (e.g., MySQL) with the **Appeal\_parcel**s table properly configured.

### PHP Script

The PHP script at the beginning of the file is responsible for handling the search functionality. It sanitizes the user input, connects to the database, and executes the appropriate query based on the input provided.

### HTML Structure

The HTML structure of the CLTIP Reconsiderations Page consists of a search form, a landowner search form, and two sections displaying the search results (pending and completed Requests for Reconsideration). Each section has a table that shows the results, and a button to export the corresponding data as a CSV file.

## Managed Forest Plan Approvers exam registration webpage:

This web page is a PHP-based exam registration form for Managed Forest Plan Approvers. It handles user input, validates and sanitizes the data, and inserts the data into a database table.

### Database Tables

The web page requires access to the following database table:

## 1. exam\_candidates

### Table Structure:

Column Name	Data Type	Description
<b>id</b>	INT	Unique identifier (Primary Key) for each candidate
<b>first_name</b>	VARCHAR	First name of the candidate
<b>last_name</b>	VARCHAR	Last name of the candidate
<b>email</b>	VARCHAR	Email address of the candidate
<b>phone</b>	VARCHAR	Phone number of the candidate
<b>sin</b>	VARCHAR	Social Insurance Number of the candidate
<b>password</b>	VARCHAR	Hashed password for the candidate's account
<b>address</b>	VARCHAR	Address of the candidate

### Constraints:

- The **id** column should be set as the primary key and auto-increment.
- The **email**, **phone**, and **sin** columns should have unique constraints to prevent duplicate entries.

### Table Creation SQL:

sqlCopy code

```
CREATE TABLE exam_candidates ( id INT AUTO_INCREMENT PRIMARY KEY, first_name VARCHAR(255) NOT NULL, last_name VARCHAR(255) NOT NULL, email VARCHAR(255) NOT NULL UNIQUE, phone VARCHAR(255) NOT NULL UNIQUE, sin VARCHAR(255) NOT NULL UNIQUE, password VARCHAR(255) NOT NULL, address VARCHAR(255) NOT NULL );
```

#### Web Page Flow:

1. Connect to the database using the **connectDB()** function from the included **library.php** file.
2. Check if the form has been submitted via a **POST** request.
3. If the form is submitted, sanitize and store the input values in variables.
4. Hash the password using the **password\_hash()** function.
5. Create an SQL query to insert the sanitized data into the **exam\_candidates** table.
6. Prepare and execute the SQL statement.
7. If the execution is successful, display a success message, otherwise, handle errors such as constraint violations for duplicate entries or other issues.
8. Render the registration form in HTML and display any success or error messages.

#### Dependencies

1. PHP server
2. Database server (e.g., MySQL)
3. **library.php** file for the **connectDB()** function

## Exporting Search Results as CSV:

Description: These PHP scripts export search results from their respectively linked pages and export them as CSV files. The exported file is named based on the table name and status (Active or Completed).

#### Flow of the Script:

1. Check if either 'export\_completed' or 'export\_pending' POST variables are set. If neither is set, terminate the script with an error message.
2. Retrieve the status ('Completed' or 'Active') based on which POST variable is set.
3. Retrieve the query string from the 'query' POST variable. If it is not set or empty, terminate the script with an error message.
4. Retrieve the table name from the 'table\_name' POST variable.
5. Connect to the database using the helper function from **../includes/library.php**.
6. Execute the query to fetch the search results.
7. Set the appropriate headers to send the CSV file as a downloadable attachment.
8. Open an output stream for writing the CSV file.
9. Write the column names to the CSV file.
10. Write the data rows to the CSV file.
11. Close the output stream and exit the script.

#### Files and Includes:

1. **../includes/library.php**: A file containing helper functions for connecting to the database.

Dependencies:

1. PHP version 7.0 or later.
2. A web server with PHP support.
3. A database server (e.g., MySQL) with the **Appeal\_parcel**s table properly configured.

## CLTIP Home:

This web page allows users to search for property, landowner, or CLTIP application information using various search criteria. It connects to a database to retrieve the results and displays them in a table.

### Database Tables

The web page accesses two database tables:

1. Parcel
2. Cltip\_application

#### Database Table: Parcel

The Parcel table contains information about properties, including the roll number, current owner, and other related details.

Field	Data Type	Description
ARN	String	Assessment Roll Number for the property
Current_owner	String	Name of the current property owner

#### Database Table: Cltip\_application

The Cltip\_application table contains information about CLTIP applications, including the PIN and other related details.

Field	Data Type	Description
-------	-----------	-------------

<b>PIN</b>	String	Personal Identification Number for the application
------------	--------	--

### PHP Code

The PHP code at the beginning of the web page handles user input and database connections. It starts by creating an array for errors and sanitizing the input data. It then connects to the database and prepares SQL queries based on the search criteria provided by the user.

### HTML Structure

The web page is divided into several sections enclosed by **<section>** tags, each with a specific purpose:

1. Property search form
2. Landowner search form
3. CLTIP application search form
4. Search results display
5. Work process links
6. Records links

### CSS

The web page uses a master.css file for styling, which is linked in the head section of the HTML file.

### Dependencies

The web page includes several dependencies in the head section of the HTML file:

1. metadata.php: Contains meta tags and other metadata for the web page.
2. header.php: Contains the header content for the web page.
3. footer.php: Contains the footer content for the web page.

## Connect with Managed Forest Plan Approvers Page:

This webpage displays a list of Managed Forest Plan Approvers and their contact information. The list is presented in a table format, showing the Name, Email, Phone, and Office Address of each approver. The page is only accessible to authenticated landowners (i.e., users who have logged in). If a user tries to access this page without being logged in, they will be redirected to the Landowner Login Page.

Language: PHP, HTML, CSS

Database tables accessed:

1. managed\_forest\_plan\_approvers

Columns:

- id (int, primary key, auto-increment)
- name (varchar)
- email (varchar)
- phone\_number (varchar)
- office\_address (text)

File structure:

/includes

- library.php
- metadata.php
- header-landowner.php
- footer.php /styles
- master.css /landowner
- connect-with-approvers.php (this file)
- landowner-login.php

External files:

1. ../includes/library.php
  - Includes the function **connectDB()** which establishes a connection to the database.
2. ../includes/metadata.php
  - Includes the metadata for the HTML head section, such as the viewport, character encoding, and other meta tags.
3. ../includes/header-landowner.php
  - Includes the header section for the landowner pages.
4. ../includes/footer.php
  - Includes the footer section for the pages.
5. /styles/master.css
  - Contains the CSS styles for the page.

Page flow:

1. The user must be logged in as a landowner. If not, they are redirected to the landowner-login.php page.
2. The database is connected to via the **connectDB()** function found in the library.php file.
3. A SQL query is executed to fetch all the rows from the **managed\_forest\_plan\_approvers** table.
4. The fetched rows are displayed in an HTML table with Name, Email, Phone, and Office Address columns.
5. The header and footer are included from external files.

Security considerations:

1. Make sure to sanitize user input and use prepared statements to prevent SQL injection attacks.
2. Ensure that the web page can only be accessed by authenticated landowners.
3. Store sensitive information, such as database credentials, in a secure configuration file.

## Conservation Land Tax Incentive Program (CLTIP) Records Search (cltip-plan, cltip-properties, cltip-landowners):

This series of webpages allows employees to search records, incorporating various bug fixes and feature enhancements. They allow users to search for properties participating in the Conservation Land Tax Incentive Program (CLTIP) using the property's ARN, the landowner's name, or the plan number. The search forms have been refined to include better input validation, error handling, and improved styling. The results are displayed in a table with a more user-friendly design, and users have the option to export the search results as a CSV file. Additionally, this includes responsive design and improved accessibility features.

### Database Tables

The web page needs to access the following database table:

1. **records**

### Database Columns

The **records** table contains information about the Conservation Land Tax Incentive Program. Based on the given code, the table is expected to have at least the following columns:

- **program**: Stores the program name (in this case, 'CLTIP').
- **ARN**: Stores the 15-digit roll number for a record.
- **Plan**: Stores the application plan number for a record.
- **Owner**: Stores the landowner's name for a record.

### PHP Code

The code uses PHP to handle user input, connect to the database, retrieve data based on the user's search criteria, and display the search results in an HTML table.

1. Sanitize and store user input.
2. Connect to the database by including `../includes/library.php` and calling `connectDB()`.
3. Depending on the input received, construct a query to search the **records** table based on the ARN, Plan, or Owner.
4. Execute the query and store the results.
5. Check if there are any rows returned by the query and store the column names if there are results.
6. Display the search results in an HTML table.



7. Provide a form to export the search results as a CSV file by submitting the data to the **export\_records.php** file.

## HTML Structure

The web page uses an HTML structure that includes several sections:

1. Header: Includes the **../includes/metadata.php** file for page metadata and the **../includes/header.php** file for the header content.
2. Search Forms: Three separate forms allow users to search by landowner's name, application plan number, or the 15-digit roll number (ARN).
3. Search Results: Displays the search results in an HTML table. If no results are found, a message is displayed to inform the user.
4. Export Button: A form that allows users to export the search results as a CSV file by submitting the data to the **export\_records.php** file.
5. Footer: Includes the **../includes/footer.php** file for the footer content.

## CSS Styles

The web page uses a stylesheet named **styles/master.css** to style the page elements.

## JavaScript Code

The JavaScript code is used to update the export form's action URL based on the user's search criteria. It also handles form submission by preventing the default behavior, updating the export form's action URL, and then submitting the form.

## Managed Forest Tax Incentive Program (MFTIP) Records Search (mftip-plan, mftip-properties, mftip-landowners):

This series of webpages allows employees to search records, incorporating various bug fixes and feature enhancements. They allow users to search for properties participating in the Managed Forest Tax Incentive Program (MFTIP) using the property's ARN, the landowner's name, or the plan number. The search forms have been refined to include better input validation, error handling, and improved styling. The results are displayed in a table with a more user-friendly design, and users have the option to export the search results as a CSV file. Additionally, this includes responsive design and improved accessibility features.

## Database Tables

The web page needs to access the following database table:

1. **records**

## Database Columns

The **records** table contains information about the Managed Forest Tax Incentive Program. Based on the given code, the table is expected to have at least the following columns:

- **program**: Stores the program name (in this case, 'MFTIP').
- **ARN**: Stores the 15-digit roll number for a record.
- **Plan**: Stores the application plan number for a record.
- **Owner**: Stores the landowner's name for a record.

## PHP Code

The code uses PHP to handle user input, connect to the database, retrieve data based on the user's search criteria, and display the search results in an HTML table.

1. Sanitize and store user input.
2. Connect to the database by including `../includes/library.php` and calling `connectDB()`.
3. Depending on the input received, construct a query to search the **records** table based on the ARN, Plan, or Owner.
4. Execute the query and store the results.
5. Check if there are any rows returned by the query and store the column names if there are results.
6. Display the search results in an HTML table.
7. Provide a form to export the search results as a CSV file by submitting the data to the `export_records.php` file.

## HTML Structure

The web page uses an HTML structure that includes several sections:

**Header:** Includes the `../includes/metadata.php` file for page metadata and the `../includes/header.php` file for the header content.

**Search Forms:** Three separate forms allow users to search by landowner's name, application plan number, or the 15-digit roll number (ARN).

**Search Results:** Displays the search results in an HTML table. If no results are found, a message is displayed to inform the user.

**Export Button:** A form that allows users to export the search results as a CSV file by submitting the data to the `export_records.php` file.

**Footer:** Includes the `../includes/footer.php` file for the footer content.

## CSS Styles

The web page uses a stylesheet named `styles/master.css` to style the page elements.

## JavaScript Code

The JavaScript code is used to update the export form's action URL based on the user's search criteria. It also handles form submission by preventing the default behavior, updating the export form's action URL, and then submitting the form.

## MNRF CLTIP & MFTIP Employee Login:

Description: This webpage serves as a login interface for MNRF staff. The user is prompted to input their roll number and password to access the system.

Database Tables it needs to authorize the Login:

1. Table Name: accounts

Column Name	Data Type	Description
rollNumber	VARCHAR	Unique identifier for each user.
pin	VARCHAR	Hashed password for each user.

1. Start the PHP script.
2. Initialize an array to store error messages.
3. Retrieve and sanitize user input for 'rollNumber' and 'pin' from the POST request.
4. Check if a 'mysitecookie' cookie exists. If so, assign its value to the 'user' variable.
5. If the form is submitted, execute the following:
  - a. Include the 'library.php' file and establish a connection to the database.
  - b. Prepare and execute an SQL query to retrieve the 'rollNumber' and 'pin' of the matching user record from the 'accounts' table.
  - c. If a matching record is found and the submitted password matches the stored password, start a session and set the 'rollNumber' session variable. Redirect the user to the 'CLTIP\_home.php' page.
  - d. If the login fails, display an error message.
  - e. If the 'remember' checkbox is selected, create a cookie named 'mysitecookie' with the roll number value and set its expiration time to one year.
6. Define the HTML structure and include necessary metadata.
7. Include the header file.

8. Create a form for the login process with the following fields:
  - a. Roll Number (text input)
  - b. Password (password input)
  - c. Remember me (checkbox)
9. Display an error message if the login fails.
10. Include the footer file.
11. Close the HTML structure.

## Landowner Login Web Page:

### Database Tables

The following database table is accessed by this web page:

1. **landowners**

#### landowners

Column	Type	Description
ARN	varchar	Account Registration Number (unique identifier)
email	varchar	Email address (unique)
password	varchar	Hashed password

### Web Page Overview

The Landowner Login web page is responsible for authenticating a landowner using either their email address or their Account Registration Number (ARN) and their password. If the credentials are valid, the user is redirected to the landowner-landing.php page. Additionally, this page supports the "remember me" functionality using cookies.

### Code Structure

1. **Sanitizing user input:** The user input for email, ARN, and password is sanitized using PHP's **filter\_var** function.

2. **Include library.php:** The **library.php** file is included, which should contain a **connectDB()** function to establish a connection with the database.
3. **Check for form submission:** The code checks if the form has been submitted using the **\$\_SERVER['REQUEST\_METHOD']** variable.
4. **Prepare and execute SQL query:** The SQL query is prepared and executed to fetch the landowner's record using either the email address or the ARN.
5. **Validate user credentials:** If a user record is found, the submitted password is compared to the password in the database. If the credentials are valid, a new session is started, and the user's ARN and email are stored in the session.
6. **Remember me functionality:** If the "remember me" checkbox is selected, the email, ARN, and password are stored in cookies. If the checkbox is not selected, any existing cookies are deleted.
7. **Redirect to the landowner home page:** If the login is successful, the user is redirected to the **landowner-landing.php** page.
8. **Display error messages:** If the login fails, an error message is displayed.

## HTML Structure

The HTML structure consists of a form containing fields for email, ARN, and password, as well as a "Forgot password" link.

## CSS

- **styles/master.css:** Contains the main CSS styles for the page.
- Google Fonts: Roboto Condensed is used as the font for the page.

## Dependencies

- PHP version 7 or later
- A PDO-compatible database such as MySQL or PostgreSQL

# MFPA Login Page:

## Description

This PHP script handles user login functionality. The script processes the login form submission, verifies the email and password against the database, and if successful, starts a session and redirects the user to the 'mfpa\_landing.php' page.

## Database Schema

The script requires access to the following database table:

**Table: exam\_candidates**

Column	Type	Description
id	INT (Primary)	Unique identifier for user
email	VARCHAR(255)	User email
password	VARCHAR(255)	Hashed user password

### File Structure

#### Files included:

1. library.php
2. metadata.php
3. header\_mfpa.php
4. footer.php
5. master.css

#### Directories included:

1. includes/
2. styles/

### Sample Request and Response

#### Request

Method: POST Content-Type: application/x-www-form-urlencoded

sqlCopy code

email=user@example.com&password=example\_password

#### Response

In case of a successful login, the user will be redirected to the 'mfpa\_landing.php' page.

In case of an unsuccessful login, an error message "Invalid email or password" will be displayed.

### Key Functions

- **connectDB**: establishes a connection to the database.
- **password\_verify**: verifies that the password matches the hashed password stored in the database.

## Logout Web Page:

Overview: The given code snippet represents a PHP-based web page that allows users to log out of an active session. When users access this web page, their session data will be cleared, and the session will be destroyed. Upon clicking the "Thank you!" button, users will be redirected to the "home.php" page.

### 1. File Structure:

- logout.php (The main file containing the code snippet)
- ../includes/metadata.php (Contains metadata for the HTML head)
- ../includes/header.php (Contains the website header)
- ../includes/footer.php (Contains the website footer)

### 3. Code Explanation:

- PHP code:
  - **session\_start();**: Initializes a new session or resumes an existing one.
  - **\$\_SESSION = array();**: Clears all session data by assigning an empty array to the **\$\_SESSION** superglobal.
  - **session\_destroy();**: Destroys the current session.
  - **if (isset(\$\_POST["submit"])) {**: Checks if the form has been submitted by verifying the "submit" POST variable's existence.
    - **header("Location: home.php");**: Redirects the user to the "home.php" page.
    - **exit();**: Terminates the script execution.
- HTML code:
  - **<!DOCTYPE html>**: Defines the document type and version of HTML.
  - **<html lang="en">**: Sets the language of the document to English.
  - **<head>**: Contains metadata and the title of the web page.
    - **<?php \$page\_title = "Logout"; ?>**: Sets the page title to "Logout".
    - **<?php include "../includes/metadata.php" ?>**: Includes the metadata file.
  - **<body class="logout">**: Defines the body content and assigns the "logout" class.
    - **<?php include '../includes/header.php'; ?>**: Includes the website header.
    - **<section>**: Contains the main content of the logout page.
      - **<form method="POST">**: Defines a form with a POST method to submit data.
        - **<h2>You have been logged out</h2>**: Displays a message informing the user that they have been logged out.
        - **<button type="submit" name="submit">Thank you!</button>**: Creates a submit button for the form.
      - **<?php include "../includes/footer.php" ?>**: Includes the website footer.
- 4. Database: The given code snippet does not interact with any database tables directly. Since the purpose of this page is to log out the user, no database access is necessary.
- 5. Dependencies:

- PHP version 5.4 or higher (for session handling)

## Annual eligible areas Conservation Land Tax Incentive Program:

This web page displays the annual eligible areas for the Conservation Land Tax Incentive Program (CLTIP) in a tabular format. The page retrieves the data from a database and offers an option to export the results as a CSV file.

### Database Tables

The web page interacts with two database tables:

1. EligibleAreaUpdates (EAU)
2. CLTIP

**Table 1: EligibleAreaUpdates (EAU)**

This table contains information about the eligible areas and their updates.

Column Name	Data Type	Description
EAU_id	INT	Primary key, unique identifier
ARN	VARCHAR	Area Registration Number
updated_at	DATE	Date of area update

**Table 2: CLTIP**

This table contains information about the Conservation Land Tax Incentive Program.

Column Name	Data Type	Description
program_id	INT	Primary key, unique identifier



<b>ARN</b>	VARCHAR	Area Registration Number
<b>program_name</b>	VARCHAR	Name of the conservation program

## Web Page Structure

The web page is divided into the following sections:

1. Metadata: Contains metadata and CSS styles for the page.
2. Header: Includes a header file for page navigation.
3. Main section: Displays the table of annual eligible areas for CLTIP.
  - If no entries are found, displays a message stating "No entries found."
  - If there are entries, displays the results in a table format.
4. Form: Allows the user to export the data as a CSV file.
5. Footer: Includes a footer file for additional information and links.

## SQL Queries

The web page uses the following SQL query to retrieve the data:

sqlCopy code

```
SELECT EAU.*, CLTIP.program_name FROM EligibleAreaUpdates EAU JOIN CLTIP ON EAU.ARN = CLTIP.ARN
WHERE YEAR(EAU.updated_at) = '$currentYear' AND CLTIP.program_name = 'CLTIP'
```

This query retrieves all the eligible areas from the **EligibleAreaUpdates** table along with the associated program name from the **CLTIP** table for the current year.

## CSV Export

The web page allows the user to export the eligible areas data as a CSV file. The **export-eligible.php** script is called when the user clicks on the "Export as CSV" button. This script processes the data and generates the CSV file for download.

## CSS Styles

The web page uses both external and internal CSS styles. The external CSS file **styles/master.css** is used for general styling, while internal CSS styles are used for the **.export-btn** class and its hover effect.

## Annual Eligible Areas Managed Forest Tax Incentive Program

This PHP web page displays a table of eligible areas for the Managed Forest Tax Incentive Program (MFTIP) based on the current year.

### Database Tables

The web page needs access to the following database tables:

1. **EligibleAreaUpdates** (alias: EAU)
2. **MFTIP**

Table: EligibleAreaUpdates (EAU)

Field	Type	Description
<b>id</b>	int	Unique identifier
<b>ARN</b>	varchar	Assessment Roll Number
<b>updated_at</b>	date	Date of the eligible area update

Table: MFTIP

Field	Type	Description
<b>id</b>	int	Unique identifier
<b>ARN</b>	varchar	Assessment Roll Number
<b>program_name</b>	varchar	Name of the Managed Forest Tax Incentive Program

### SQL Queries

QueryYear

The following SQL query retrieves the eligible areas for the MFTIP in the current year.

sqlCopy code

```
SELECT EAU.*, MFTIP.program_name FROM EligibleAreaUpdates EAU JOIN MFTIP ON EAU.ARN = MFTIP.ARN WHERE YEAR(EAU.updated_at) = '$currentYear' AND MFTIP.program_name = 'MFTIP'
```

### Page Structure

1. Metadata and stylesheets are included in the **<head>** section.
2. Header with navigation is included using **<?php include '../includes/header.php'; ?>**.
3. Main content is wrapped in a **<section>** element with a **.section-box** class.
4. If there are no eligible areas for the current year, a "No entries found." message is displayed.
5. If there are eligible areas, a table is generated to display the information.
6. An export button is provided to download the eligible areas data as a CSV file.

### CSS

The CSS for this page is primarily located in the **styles/master.css** file. Additional styling specific to the export button is included in the **<style>** section of the web page.

## Forgot Password Webpage:

### Overview

This web page allows users to reset their password by providing their email address, a new password, and a confirmation for the new password. The page is written in PHP and connects to a MySQL database to verify and update the user's credentials.

### Database Tables

The script accesses the following database table:

#### Table: users

Field	Type	Description
id	INT	The primary key and unique ID of the user
email	VARCHAR(255)	The user's email address

<b>password</b>	VARCHAR(255)	The hashed password of the user
-----------------	--------------	---------------------------------

### PHP Script

1. The script imports the **library.php** file which provides the **connectDB()** function to connect to the database.
2. It initializes an array, **\$errors**, to store any errors that may occur during the password reset process.
3. The script sanitizes user input from the **\$\_POST** superglobal for **email**, **password**, and **confirmPassword**.
4. If the form is submitted, the script connects to the database and selects the **email** and **password** columns from the **users** table for the provided email address.
5. It validates the email address and checks if it exists in the database.
6. If the email address is valid and exists in the database, it proceeds to validate the new password based on specific criteria:
  - At least one uppercase letter
  - At least one lowercase letter
  - At least one number
  - At least one special character
  - Minimum length of 8 characters
7. If the new password is valid, it checks if the confirmed password matches the new password.
8. If there are no errors, the new password is hashed and updated in the **users** table for the provided email address.
9. The user is then redirected to the **home.php** page.

### HTML Structure

The HTML structure of the page includes the following:

- Metadata and external stylesheet **master.css** are imported.
- The header is included from **header.php**.
- A form section is provided for the user to enter their email address, new password, and confirmation password.
  - Error messages are displayed based on specific error conditions.
- The footer is included from **footer.php**.

### Dependencies

- PHP 7.4 or later
- MySQL Database
- **library.php** with the **connectDB()** function
- **header.php**
- **footer.php**

- **master.css** stylesheet

## Find a Managed Forest Plan Approver Webpage:

This web page is designed to search and display Managed Forest Plan Approvers based on the user's input. The user can search either by Landowner's name and organization or by the Municipality's 15-digit roll number. The search results are displayed in a table format.

### Database Tables

The following database tables are accessed in this web page:

1. **accounts**
2. **Municipality**
3. **Parcel**

### Technical Documentation

#### PHP Code

1. The code initializes an array called **\$errors** to store errors.
2. The code sanitizes and stores user input from the **\$\_POST** array into local variables.
3. The code includes a PHP file, **library.php**, to connect to the database using a function called **connectDB()**.
4. The code prepares a query based on which search button (**searchLO** or **searchMunicipality**) is clicked. If no button is clicked, it defaults to selecting data from the **Parcel** table.
5. The code executes the prepared query and fetches the results as an associative array.
6. The code retrieves the column names from the query result.

#### HTML Structure

1. The **<!DOCTYPE html>** declaration defines the document type and version of HTML.
2. The **<html>** element is the root element of the page and has a **lang** attribute set to "en" for English.
3. The **<head>** element contains metadata, including the **\$page\_title** variable set to "Managed Forest Plan" and a PHP include for additional metadata.
4. The **<body>** element contains the content of the document.
5. The PHP include for the header is added.
6. A **<form>** element is used with a **POST** method for submitting user input.
7. The form contains two search sections: one for searching by Landowner and one for searching by Municipality.
8. The search results are displayed in a **<table>** element with column headers and table data.
9. The PHP include for the footer is added.

#### CSS Classes

1. **.queryForm** - Styles the form containing the search input and results.
2. **.search** - Styles the search input section of the form.
3. **.searchResults** - Styles the search results section of the form.
4. **.results** - Styles the container for the results table.
5. **.resultsTable** - Styles the table displaying search results.

### Dependencies

1. PHP 7 or higher
2. Database connection (using **library.php** and **connectDB()** function)

### Security

1. The code uses **trim()** and **filter\_var()** functions to sanitize user inputs before using them in queries or displaying them in the form.

## Find a Landowner in Managed Forest Plan Webpage:

This web page is a search interface for a Managed Forest Plan. It allows users to search for landowners and their properties using various search criteria such as landowner name, plan number, or roll number. The search results are displayed in a tabular format.

### Database Tables

The PHP script in this web page accesses the following database tables:

1. **accounts**
2. **Parcel**
3. **Municipality**

### Code Walkthrough

#### Sanitizing Input

The script starts by creating an empty array **\$errors** to store any errors that might occur. Then, it sanitizes the user inputs using **trim()** and **filter\_var()** functions. The sanitized inputs are stored in the following variables:

- **\$planNumber**
- **\$property**
- **\$LOFirstName**
- **\$LOLastName**
- **\$LOOrg**

#### Database Connection

The script then connects to the database by including a **library.php** file and calling the **connectDB()** function. The function returns a PDO object that is stored in the **\$pdo** variable.

### Query Preparation

The script checks which search criteria has been submitted by the user and prepares the corresponding query:

- If the user submitted a search by plan number, the query selects all rows from the **accounts** table.
- If the user submitted a search by property, the query selects all rows from the **Parcel** table.
- If the user submitted a search by landowner, the query selects all rows from the **Municipality** table.
- If none of the above conditions are met, the query selects all rows from the **Parcel** table as the default view.

### Query Execution

The script then executes the prepared query using a PDO prepared statement. It fetches all the rows from the executed query and stores them in the **\$row** variable. It also retrieves the column names of the fetched rows and stores them in the **\$columnNames** variable.

### HTML Output

The HTML portion of the script includes a form element that contains the search fields and buttons for the user to input their search criteria. The search results are displayed in a table within a **searchResults** div. The table's column headers are generated using the **\$columnNames** variable, and the table's row data is populated using the **\$row** variable.

### File Structure

The web page uses the following file structure:

- **library.php**: A PHP file that contains the **connectDB()** function for establishing a connection to the database.
- **metadata.php**: A PHP file that contains metadata for the web page.
- **header.php**: A PHP file that contains the header of the web page.
- **footer.php**: A PHP file that contains the footer of the web page.

## Lists to Send to MPAC Webpage:

This web page displays two tables containing data from the "plans2" database table, one showing results for the current year and another for the current month. Additionally, the page provides buttons to export the data as CSV files.

Database Table:

- plans2

Fields in the plans2 table:

1. EA Approved On

PHP Dependencies:

- ../includes/library.php
- ../includes/metadata.php
- ../includes/header.php
- ../includes/footer.php

CSS Dependencies:

- styles/master.css

Database Connections:

1. Connect to the database using the connectDB() function from the library.php file.
2. Prepare and execute two SELECT queries to retrieve data for the current year and current month.

Data Flow:

1. Get the current year and month values.
2. Connect to the database.
3. Execute the SELECT queries to fetch data for the current year and month.
4. Check if there are any rows returned by the queries.
5. Fetch all the rows and store them in arrays.
6. Get all the column names of the fetched rows.
7. Display the data in HTML tables.
8. Include buttons to export the data as CSV files.

HTML Structure:

- <!DOCTYPE html>
- <html lang="en"> - <head> - Page title and metadata - Stylesheets and inline styles - <body> - Header - <section class="section-box"> (List from the Current Year to Send to MPAC) - Table with results from the current year - Export button and form for the current year - <section class="section-box"> (List from the Current Month to Send to MPAC) - Table with results from the current month - Export button and form for the current month - Footer

Export Functionality:

1. Include a form with a hidden input field for the table name, current year, and current month.
2. Include a submit button with the class "export-btn".
3. Set the form's action attribute to "export\_to\_csv.php" and the method attribute to "post".



Styles:

- .export-btn: Styles for the export button, including background color, text color, font size, padding, margin, cursor, border-radius, box-shadow, and hover effect.

## Managed Forest Plan:

Description: This web page allows users to search for managed forest plans by plan number, property, or landowner. Users can input a 5-digit plan number, a 15-digit roll number for a property, or the first name, last name, or organization of a landowner. The search results are displayed in a table.

### Database Tables Accessed:

1. Parcel
2. Landowners
3. Accounts
4. Municipality

### PHP Code Overview:

1. Define an array for storing errors.
2. Get and sanitize user input from the form.
3. Connect to the database.
4. Based on the input provided, execute the appropriate query to fetch data from the database.
5. Fetch all the data from the query result as an associative array.
6. Get column names from the query result.
7. If the search result is empty, fetch data from the Accounts table as the default view.

### HTML Structure:

1. Include metadata and header files.
2. Create a form with three sections: Plans, Property, and Landowner.
3. In each section, provide an input field for the user to enter the respective search criteria.
4. Display search results in a table format.

### Database Schema:

1. Parcel table:

Column Name	Data Type	Description
PYP	Integer	Plan Year Plan (5-digit plan number)

2. Landowners table:

Column Name	Data Type	Description
<b>ARN</b>	Integer	Roll Number (15-digit roll number)

3. Accounts table:

Column Name	Data Type	Description
<b>rollNumber</b>	Integer	Roll Number (15-digit roll number)
<b>firstName</b>	String	Landowner's first name
<b>lastName</b>	String	Landowner's last name

4. Municipality table:

Column Name	Data Type	Description
<b>name</b>	String	Municipality or organization's name

#### Dependencies:

1. library.php: Contains the function to connect to the database.
2. metadata.php: Contains metadata for the web page.
3. header.php: Contains the header of the web page.
4. footer.php: Contains the footer of the web page.

## Core Functionalities:

### 1. User Authentication

- Secure login for Landowners, MFPA's, MFTIP, and CLTIP employees using username, password or other varying parameters.

### 2. Landowner Application Management

- Landowners can securely access their plan details and 5-year progress reports
- MFTIP and CLTIP employees can update Landowner application status
- Landowners can check when their application status changes
- MFPA's can submit managed forest plan applications
- Data validation for uploaded information
- Flag and send changes to property information to MFTIP and CLTIP employees monthly

### **3. MPAC Integration and Updates**

- Import data provided by MPAC into the database
- Export data to send MPAC updates about changes to properties' participation status and area for the current and previous tax years
- Export data to send MPAC annual lists (new, renewals, removals) of newly participating and newly non-participating properties for the following tax year

### **4. Database Management and Administration**

- MFTIP and CLTIP employees can make updates to individual records in the database
- MNRF IT department can make changes to the Database to push updates or resolve issues

### **5. MFPA Exam Registration and Study Materials**

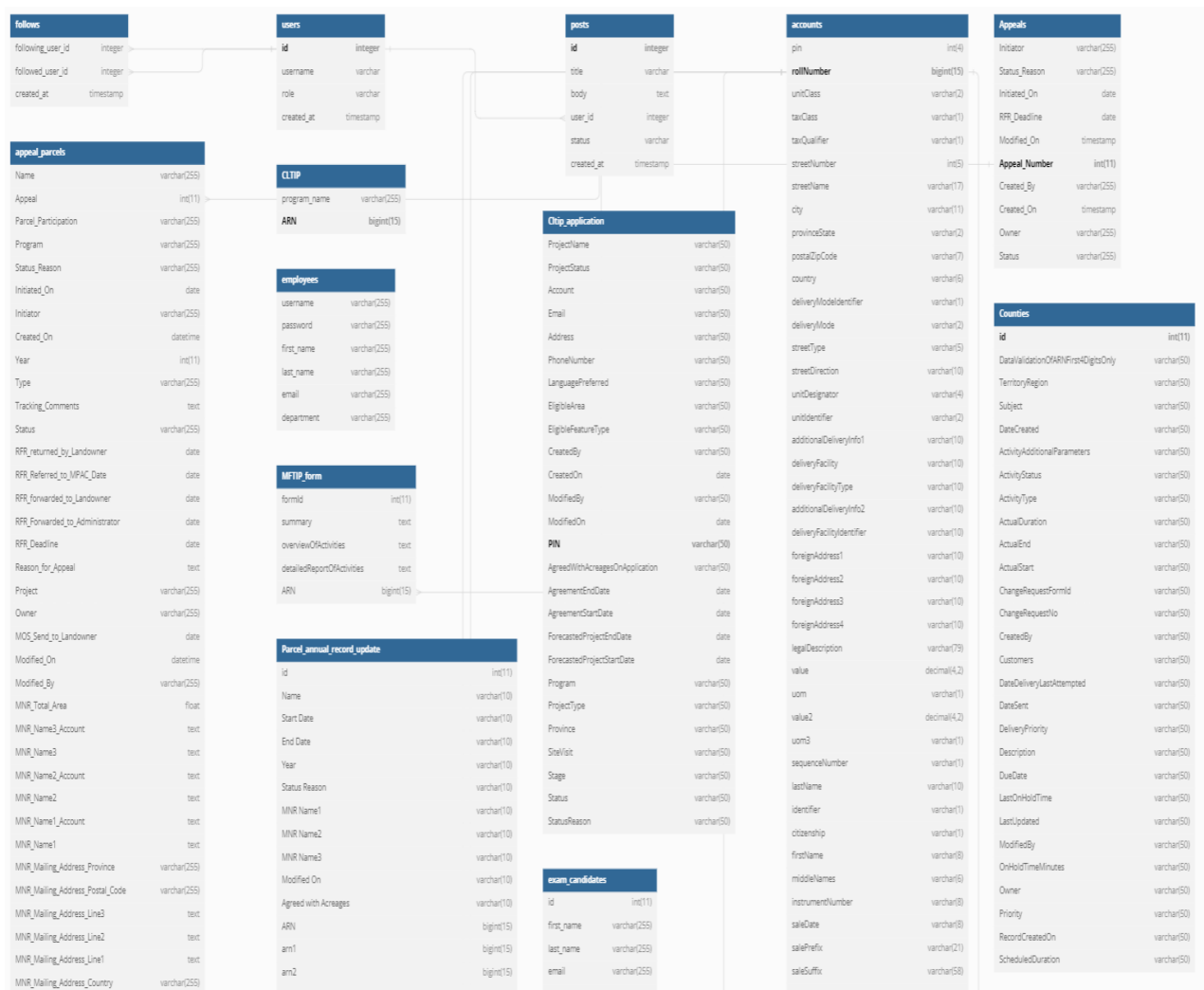
- System allows registration for MFPA exam
- Provide registered candidates access to exam information and study materials
- User Authentication

### **6. Exporting Search Results:**

- MFTIP and CLTIP employees can export the search results from the database in a csv format.

## Database Schema:

Due to the large data tables, a full database schema is attached separately. The database schema below is a continuous image, however the tables have been cut off due to the limitation size of the document.



```

    erDiagram
        EligibleAreaUpdates ||--o{ Municipality : "has"
        EligibleAreaUpdates ||--o{ ParcelAnnualRecord : "has"
        EligibleAreaUpdates ||--o{ Landowners : "has"
        EligibleAreaUpdates ||--o{ ManagedForestPlanApprovers : "has"
        EligibleAreaUpdates ||--o{ Parcels : "has"
        Municipality ||--o{ ParcelAnnualRecord : "has"
        Municipality ||--o{ Landowners : "has"
        Municipality ||--o{ ManagedForestPlanApprovers : "has"
        Municipality ||--o{ Parcels : "has"
        ParcelAnnualRecord ||--o{ Landowners : "has"
        ParcelAnnualRecord ||--o{ ManagedForestPlanApprovers : "has"
        ParcelAnnualRecord ||--o{ Parcels : "has"
        Landowners ||--o{ ManagedForestPlanApprovers : "has"
        Landowners ||--o{ Parcels : "has"
        ManagedForestPlanApprovers ||--o{ Parcels : "has"
  
```

**EligibleAreaUpdates**

- id (PK)
- ARN (FK)
- updated\_at
- Area\_id (FK)

**MFTIP**

- program\_name
- ARN (FK)

**ParcelAnnualRecord**

- id (PK)
- Parcel Year Participation
- Name
- Start Date
- End Date
- Year
- Status Reason
- MNR Name1
- MNR Name2
- MNR Name3
- Modified On
- Agreed with Acreages
- ARN (FK)
- am1
- am2
- am3
- am4
- Building on CL
- CC.Subtype #1 Natural Heritage Area
- CC.Subtype #10 Municipally designated lands
- CC.Subtype #11 Great Lakes Wetlands
- CC.Subtype #2 Regionally significant AVS
- CC.Subtype #3 Species occurrence
- CC.Subtype #4 Habitat of Special Concern species
- CC.Subtype #5 OLL Featured Area
- CC.Subtype #6 Abutting within Provincial Park
- CC.Subtype #7 Oak Ridges Moraine natural core
- CC.Subtype #8 Niagara Escarpment Area
- CC.Subtype #9 Regional or watershed plan
- CL Confirmation Number
- CL Eligible Area overlaps with MF Eligible Area
- CL Feature Area Names
- CL Inquiry
- CL Type #1
- Comments
- County
- Created By
- Created On
- dateleep/MHAddressChecked
- Email Address
- Error Log
- GS Verification Results
- GS Verification Status
- GS Verification Year
- Harvesting Activity Commercial Intent
- Harvesting Activity Personal Intent
- Include in MPAC Extract
- Linked Parcel Year Participation
- Mismatch Comments
- Mismatch Comments Previous
- Mismatch Eligible Area
- Mismatch Eligible Area Previous
- Mismatch Landowner
- Mismatch Landowner Previous
- Mismatch Legal Description
- Mismatch Legal Description Previous
- Mismatch Mailing Address
- Mismatch Mailing Address Previous
- Mismatch Total Area
- Mismatch Total Area Previous
- MNR Address
- MNR City
- MNR Eligible Area
- MNR Legal Description
- MNR Mailing Address Country
- MNR Mailing Address Line2

**Municipality**

- Name
- County
- Created By
- Created On
- Full Name
- Modified By
- Modified On
- Municipality Code
- Status
- Status Reason
- ARN (FK)

**Parcel record**

- id (PK)
- Start Date
- End Date
- Year
- Status Reason
- MNR Name1
- MNR Name2
- MNR Name3
- Modified On
- Agreed with Acreages
- ARN (FK)
- am1
- am2
- am3
- am4
- Building on CL
- CC.Subtype\_1\_Natural\_Heritage\_Area
- CC.Subtype\_10\_Municipally\_designated\_lands
- CC.Subtype\_11\_Great\_Lakes\_Wetlands
- CC.Subtype\_2\_Regionally\_significant\_AVS
- CC.Subtype\_3\_Species\_occurrence
- CC.Subtype\_4\_Habitat\_of\_Special\_Concern\_species
- CC.Subtype\_5\_OLL\_Featured\_Area
- CC.Subtype\_6\_Abutting\_within\_Provincial\_Park
- CC.Subtype\_7\_Oak\_Ridges\_Moraine\_natural\_core
- CC.Subtype\_8\_Niagara\_Escarpment\_Area
- CC.Subtype\_9\_Regional\_or\_watershed\_plan
- CL Confirmation Number
- CL\_Eligible\_Area\_overlaps\_with\_MF\_Eligible\_Area
- CL\_Feature\_Area\_Names
- CL\_Inquiry

**Landowners**

- ARN (FK)
- propo\_info
- plan\_number
- approved\_by
- plan\_start\_date
- plan\_end\_date
- progress\_report\_due
- renewal\_plan\_due
- email

**Managed forest plan approvers**

- id (PK)
- name
- email
- phone\_number
- office\_address

**Parcels**

- id (PK)
- ARN (FK)
- Current\_owner
- Current\_eligible\_area
- PIP
- link to annual participation information
- current\_MFTIP\_plan
- current CLTP app
- past MFTIP plans
- past CLTP apps
- Term end date
- Data validation should ensure that CL area (if participating) +

CCLSubtype #5 OLL Featured Area	varchar(10)	MNR Mailing Address Line3	varchar(10)	CL_Type_1	text	password	varchar(255)
CCLSubtype #6 Abutting/ within Provincial Park	varchar(10)	MNR Mailing Address Province	varchar(10)	Comments	text	status_for_tax_year	varchar(255)
CCLSubtype #7 Oak Ridges Moraine natural core	varchar(10)	MNR Name1 Account	varchar(10)	County	text	first_name	varchar(255)
CCLSubtype #8 Niagara Escarpment Area	varchar(10)	MNR Name1 Citizenship	varchar(10)	Created_By	text	last_name	varchar(255)
CCLSubtype #9 Regional or watershed plan	varchar(10)	MNR Name1 Type	varchar(10)	Created_On	varchar(255)		
CL Confirmation Number	varchar(10)	MNR Name2 Account	varchar(10)	dateKeepMNRAddressChecked	varchar(255)		
CL Eligible Area overlaps with MF Eligible Area	varchar(10)	MNR Name2 Citizenship	varchar(10)	E_mail_Address	text		
CL Feature Area Names	varchar(10)	MNR Name2 Type	varchar(10)	Error_Log	text		
CL Inquiry	varchar(10)	MNR Name3 Account	varchar(10)	GIS_Verification_Results	text		
CL Type #1	varchar(10)	MNR Name3 Citizenship	varchar(10)	GIS_Verification_Status	text		
Comments	varchar(10)	MNR Name3 Type	varchar(10)	GIS_Verification_Year	int(11)		
County	varchar(10)	MNR Postal Code	varchar(10)	Harvesting_Activity_Commercial_Intent	text		
Created By	varchar(10)	MNR Total Area	varchar(10)	Harvesting_Activity_Personal_Intent	text		
Created On	varchar(10)	mnr_FirstName1	varchar(10)	Include_in_MFAC_Extract	varchar(255)		
dateKeepMNRAddressChecked	varchar(10)	mnr_FirstName2	varchar(10)	Linked_Parcel_Year_Participation	varchar(255)		
E-mail Address	varchar(10)	mnr_LastName1	varchar(10)	Mismatch_Comments	text		
Error Log	varchar(10)	mnr_LastName2	varchar(10)	Mismatch_Comments_Previous	text		
GIS Verification Results	varchar(10)	mnr_MiddleName1	varchar(10)	Mismatch_Eligible_Area	float		
GIS Verification Status	varchar(10)	mnr_MiddleName2	varchar(10)	Mismatch_Eligible_Area_Previous	float		
GIS Verification Year	varchar(10)	Modified By	varchar(10)	Mismatch_Landowner	text		
Harvesting Activity Commercial Intent	varchar(10)	MPAC Eligible Area	varchar(10)	Mismatch_Landowner_Previous	text		
Harvesting Activity Personal Intent	varchar(10)	MPAC Legal Description	varchar(10)	Mismatch_Legal_Description	text		
Include in MFAC Extract	varchar(10)	MPAC Mailing Address City	varchar(10)	Mismatch_Legal_Description_Previous	text		
Linked Parcel Year Participation	varchar(10)	MPAC Mailing Address Country	varchar(10)	Mismatch_Mailing_Address	text		
Mismatch Comments	varchar(10)	MPAC Mailing Address Line1	varchar(10)	Mismatch_Mailing_Address_Previous	text		
Mismatch Comments Previous	varchar(10)	MPAC Mailing Address Line2	varchar(10)	Mismatch_Total_Area	float		
Mismatch Eligible Area	varchar(10)	MPAC Mailing Address Line3	varchar(10)	Mismatch_Total_Area_Previous	float		
Mismatch Eligible Area Previous	varchar(10)	MPAC Mailing Address Postal Code	varchar(10)	MNR_Address	text		
Mismatch Landowner	varchar(10)	MPAC Mailing Address Province	varchar(10)	MNR_City	text		
Mismatch Landowner Previous	varchar(10)	MPAC Name1	varchar(10)	MNR_Eligible_Area	float		
Mismatch Legal Description	varchar(10)	MPAC Name1 Citizenship	varchar(10)	MNR_Legal_Description	text		
Mismatch Legal Description Previous	varchar(10)	MPAC Name1 Type	varchar(10)	MNR_Mailing_Address_Country	text		
Mismatch Mailing Address	varchar(10)	MPAC Name2	varchar(10)	MNR_Mailing_Address_Line2	text		
Mismatch Mailing Address Previous	varchar(10)	MPAC Name2 Citizenship	varchar(10)	MNR_Mailing_Address_Line3	text		
Mismatch Total Area	varchar(10)	MPAC Name2 Type	varchar(10)	MNR_Mailing_Address_Province	text		
Mismatch Total Area Previous	varchar(10)	MPAC Name3	varchar(10)	MNR_Name1_Account	text		
MNR Address	varchar(10)	MPAC Name3 Citizenship	varchar(10)	MNR_Name1_Citizenship	text		
MNR City	varchar(10)	MPAC Name3 Type	varchar(10)	MNR_Name1_Type	text		
MNR Eligible Area	varchar(10)	MPAC Total Area	varchar(10)	MNR_Name2_Account	text		
MNR Legal Description	varchar(10)	MPAC Verification Status	varchar(10)	MNR_Name2_Citizenship	text		
MNR Mailing Address Country	varchar(10)	Municipality	varchar(10)	MNR_Name2_Type	text		
MNR Mailing Address Line2	varchar(10)	Net Eligible Forest Area (D)	varchar(10)	MNR_Name3_Account	text		
MNR Mailing Address Line3	varchar(10)	NotOverrideMNRData	varchar(10)	MNR_Name3_Citizenship	text		
MNR Mailing Address Province	varchar(10)	Other Lands (I)	varchar(10)	MNR_Name3_Type	text		
MNR Name1 Account	varchar(10)	Owner	varchar(10)	MNR_Phone_Number	varchar(255)		
MNR Name1 Citizenship	varchar(10)	Parcel	varchar(10)	MNR_Postal_Code	varchar(255)		
MNR Name1 Type	varchar(10)	Parcel involved in split/consol?	varchar(10)	MNR_Province	text		
MNR Name2 Account	varchar(10)	Participate FFCTRP Before	varchar(10)	MNR_Street_Number	text		
MNR Name2 Citizenship	varchar(10)	Participate FFCTRP Now	varchar(10)	MNR_Total_Area	float		
MNR Name2 Type	varchar(10)	Participate MFTIP Now	varchar(10)	MPAC_Address	text		
MNR Name3 Account	varchar(10)	Participated MFTIP before?	varchar(10)	MPAC_City	text		
MNR Name3 Citizenship	varchar(10)	Participation Status	varchar(10)	MPAC_Eligible_Area	float		

Ownership\_change

id	int(11)
Project Name	varchar(50)
Project Type	varchar(50)
Account	varchar(50)
Email	varchar(50)
Phone Number	varchar(50)
New Account	varchar(50)
New Email	varchar(50)
New Phone Number	varchar(50)
Project Start Date	date
Project End Date	date
Ownership Change Date	date
Created By	varchar(50)
Created On	date
Modified By	varchar(50)
Modified On	date
PIN	varchar(50)
ARN	bigint(15)

plans

id	int(11)
Aboriginal Groups Involved	varchar(100)
Animal Care Protocol	varchar(100)
Approach or Methodology	varchar(100)
Approval Comments	varchar(255)
Associated Parcels	int(11)
Benefits	varchar(255)
Case	varchar(20)
Census Subdivision	varchar(100)
City	varchar(100)
Claims Override Reason	varchar(255)
Client Share (Base)	decimal(10,2)
Client Share %	decimal(5,2)
Client Share Amount	decimal(10,2)
Client Type for HST	varchar(100)
Closure Comments	varchar(255)
Closure Reason	varchar(100)
Community Support	varchar(100)
Construction End Date	date
Construction Start Date	date
Costs Incurred From	date
Costs Incurred To	date
County	varchar(100)
Created By (Delegate)	varchar(100)
Currency	varchar(10)

MNR Name3 Type	varchar(10)	Previous Parcel Year Participation	varchar(10)	MPAC_Legal_Description	text	currentnoyTemp	decimal(10,2)
MNR Postal Code	varchar(10)	Previous Reason	varchar(10)	MPAC_Mailing_Address_Country	text	currentnoyTemp (Base)	decimal(10,2)
MNR Total Area	varchar(10)	Previous Status Reason	varchar(10)	MPAC_Mailing_Address_Line2	text	Description	text
mnr_FirstName1	varchar(10)	Program	varchar(10)	MPAC_Mailing_Address_Line3	text	Duty To Consult	varchar(100)
mnr_FirstName2	varchar(10)	Project	varchar(10)	MPAC_Mailing_Address_Province	text	EA Approved On	date
mnr_LastName1	varchar(10)	Property is eligible under CLTIP	varchar(10)	MPAC_Name1_Account	text	Exchange Rate	decimal(10,4)
mnr_LastName2	varchar(10)	Provided On	varchar(10)	MPAC_Name1_Citizenship	text	Executed Date	date
mnr_MiddleName1	varchar(10)	Reason	varchar(10)	MPAC_Name1_Type	text	Federal Riding	varchar(100)
mnr_MiddleName2	varchar(10)	Reason for MPAC Extract	varchar(10)	MPAC_Name2_Account	text	Federal Share %	decimal(5,2)
Modified By	varchar(10)	Sent to MPAC On Date	varchar(10)	MPAC_Name2_Citizenship	text	Federal Share Amount	decimal(10,2)
MPAC Eligible Area	varchar(10)	Special Policy	varchar(10)	MPAC_Name2_Type	text	Federal Share Amount (Base)	decimal(10,2)
MPAC Legal Description	varchar(10)	Status	varchar(10)	MPAC_Name2_Account	text	Final Report Date	date
MPAC Mailing Address City	varchar(10)	Submitted Eligible Area	varchar(10)	MPAC_Name2_Citizenship	text	Final Report Date Multiyear	date
MPAC Mailing Address Country	varchar(10)	Temp CLConfirmationNumber	varchar(10)	MPAC_Name2_Type	text	First Contract Award	date
MPAC Mailing Address Line1	varchar(10)	Temporary ID	varchar(10)	MPAC_Phone_Number	varchar(255)	First Tender Call	date
MPAC Mailing Address Line2	varchar(10)	Total Conservation Lands (H)	varchar(10)	MPAC_Postal_Code	varchar(255)	Forecasted Construction End Date	date
MPAC Mailing Address Line3	varchar(10)	Total Elig. Forest Area (G)	varchar(10)	MPAC_Province	text	Forecasted Construction Start Date	date
MPAC Mailing Address Postal Code	varchar(10)	Total Managed Forest Area (B)	varchar(10)	MPAC_Street_Number	text	Project Types Import Id	int(11)
MPAC Mailing Address Province	varchar(10)	Total Open Area (E)	varchar(10)	MPAC_Total_Area	float		
MPAC Name1	varchar(10)	Total Open Area No trees (F)	varchar(10)	Original_Data_Entry_Date	varchar(255)		
MPAC Name1 Citizenship	varchar(10)	Total Other Lands (J)	varchar(10)	Parcel_Identifier	varchar(255)		
MPAC Name1 Type	varchar(10)	Total Residence Area (I)	varchar(10)	Parcel_Identifier_Previous	varchar(255)		
MPAC Name2	varchar(10)	Total Summary	varchar(10)	Parcel_Status	text		
MPAC Name2 Citizenship	varchar(10)	Update Address	varchar(10)	Parcel_Status_Previous	text		
MPAC Name2 Type	varchar(10)	Update Landowners	varchar(10)	Region	text		
MPAC Name3	varchar(10)	a_record_id	int(11)	Submitted_By	text		
MPAC Name3 Citizenship	varchar(10)			Surveyor_Name	text		
MPAC Name3 Type	varchar(10)			Township	text		
MPAC_total_area	varchar(10)			Updated_By	text		
MPAC Verification Status	varchar(10)			Updated_On	varchar(255)		
Municipality	varchar(10)			Verification_Method	text		
Net Eligible Forest Area (D)	varchar(10)			Verification_Method_Previous	text		
NotOverrideMNRData	varchar(10)			Work_Order_Number	varchar(255)		
Other Lands (I)	varchar(10)						
Owner	varchar(10)						
Parcel	varchar(10)						
Parcel Involved in split/consol?	varchar(10)						
Participate FPCTRP Before	varchar(10)						
Participate FPCTRP Now	varchar(10)						
Participate MFTIP Now	varchar(10)						
Participated MFTIP before?	varchar(10)						
Participation Status	varchar(10)						
Previous Parcel Year Participation	varchar(10)						
Previous Reason	varchar(10)						
Previous Status Reason	varchar(10)						
Program	varchar(10)						
Project	varchar(10)						
Property is eligible under CLTIP	varchar(10)						
Provided On	varchar(10)						
Reason	varchar(10)						
Reason for MPAC Extract	varchar(10)						
Sent to MPAC On Date	varchar(10)						
Special Policy	varchar(10)						
Status	varchar(10)						

Send to MPAC

Project Name	int(11)
Project Type	int(11)
Account	int(11)
Project Start Date	int(11)
Project End Date	int(11)
Created By	text
Created On	date
Modified By	text
Modified On	date
Status	text
Status Type	text
Date Sent	date
Reason Sent	text
MPAC_id	int(11)

Technical reviews

id	bigint(11)
Name	varchar(10)
Individual Review Type	varchar(10)
Due Date	varchar(10)
Submitted On	varchar(10)
Owner	varchar(10)
Total Score	varchar(10)
Status Reason	varchar(10)
Modified On	varchar(10)
20 Year Plan Dates Correct	varchar(10)
Approved By	varchar(10)
Approved On	varchar(10)
Assessment Form Id	varchar(10)
Assessment Package Number	varchar(10)
Client Lead	varchar(10)
Comments	varchar(10)
Created By	varchar(10)
Created On	varchar(10)

Submitted Eligible Area	varchar(10)
Temp CLConfirmationNumber	varchar(10)
Temporary ID	varchar(10)
Total Conservation Lands (H)	varchar(10)
Total Elig. Forest Area (G)	varchar(10)
Total Managed Forest Area (B)	varchar(10)
Total Open Area (E)	varchar(10)
Total Open Area No trees (F)	varchar(10)
Total Other Lands (I)	varchar(10)
Total Residence Area (C)	varchar(10)
Total Summary	varchar(10)
Update Address	varchar(10)
Update Landowners	varchar(10)

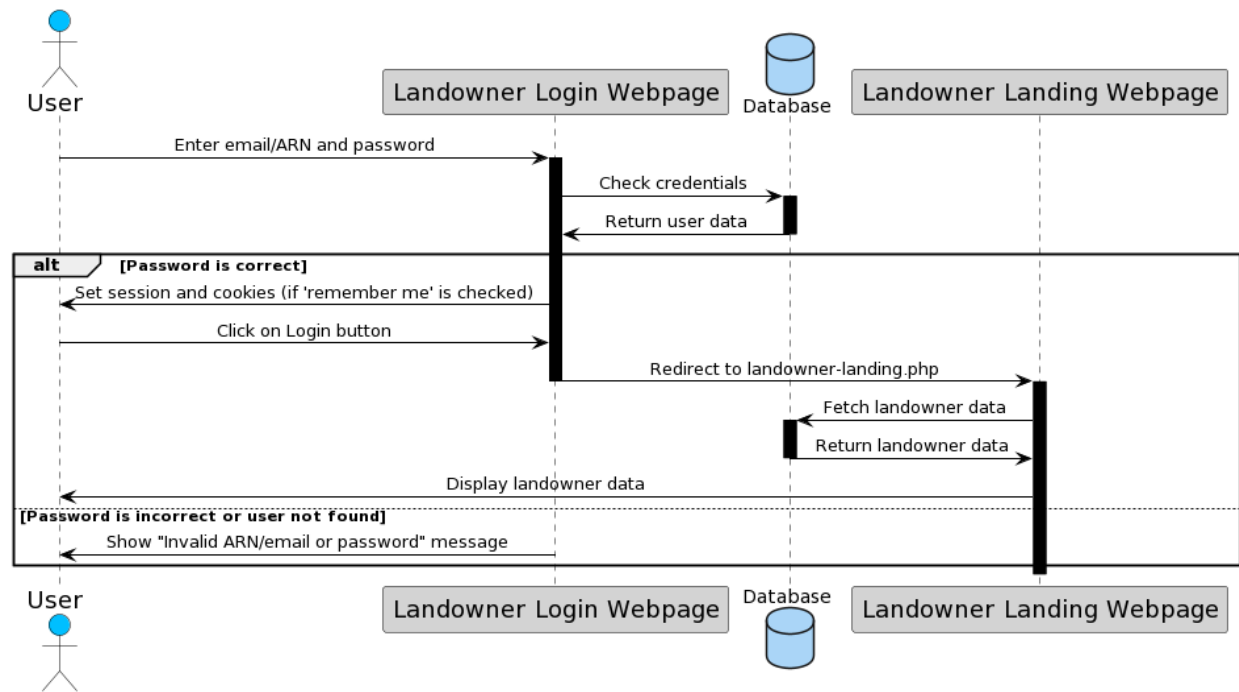
records	
id	int(11)
ARN	bigint(15)
Program	records_program_enum
Plan	int(11)
PlanStatus	varchar(255)
PlanTerm	varchar(255)
Owner	varchar(255)
PropertiesInPlan	text

Currency	varchar(10)
Evaluation	varchar(10)
Exchange Rate	varchar(10)
Mandatory Items Score	varchar(10)
Maximum Score	varchar(10)
Meets Min Standards	varchar(10)
MFPA	varchar(10)
Modified By	varchar(10)
Overall Quality Comments	varchar(10)
Overall Quality Score 1	varchar(10)
Overall Quality Score 2	varchar(10)
Overall Quality Score 3	varchar(10)
Overall Quality Score 4	varchar(10)
Overall Quality Score 5	varchar(10)
Overall Quality Score 6	varchar(10)
Portal User	varchar(10)
Recommendation	varchar(10)
Section 3.2 Score	varchar(10)
Section 1 Comments	varchar(10)
Section 1.1 Score	varchar(10)
Section 1.2 Score	varchar(10)
Section 10 Comments	varchar(10)
Section 10 Score	varchar(10)
Section 2 Comments	varchar(10)
Section 2 Score	varchar(10)
Section 2 Score in %	varchar(10)
Section 2.1 Score	varchar(10)
Section 2.2 Score	varchar(10)
Section 3 Comments	varchar(10)
Section 3.1 Score	varchar(10)
Section 4 Comments	varchar(10)
Section 4 Score	varchar(10)
Section 5.1 Score	varchar(10)
Section 5.2 Score	varchar(10)
Section 5.3 Score	varchar(10)
Section 6.1 Score	varchar(10)
Section 6.2 Score	varchar(10)
Section 7 Comments	varchar(10)
Section 7 Eligibility Score	varchar(10)
Section 7.1 Score	varchar(10)
Section 7.2 Score	varchar(10)
Section 7.3 Score	varchar(10)
Section 7.4 Score	varchar(10)
Section 7.5 Score	varchar(10)
Section 7.6 Score	varchar(10)
Section 8 Comments	varchar(10)
Section 8 Score	varchar(10)
Section 9 Comments	varchar(10)
Section 9 Score	varchar(10)
Status	varchar(10)
Total Percentage	varchar(10)
plan_id	int(11)

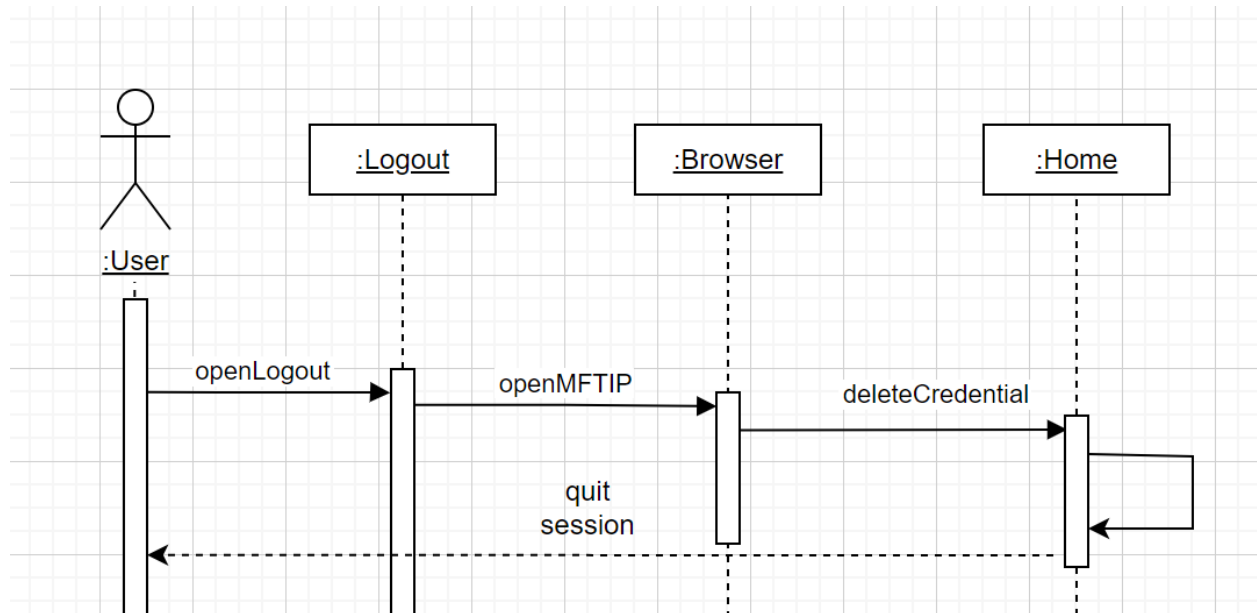


## Sequence Diagrams:

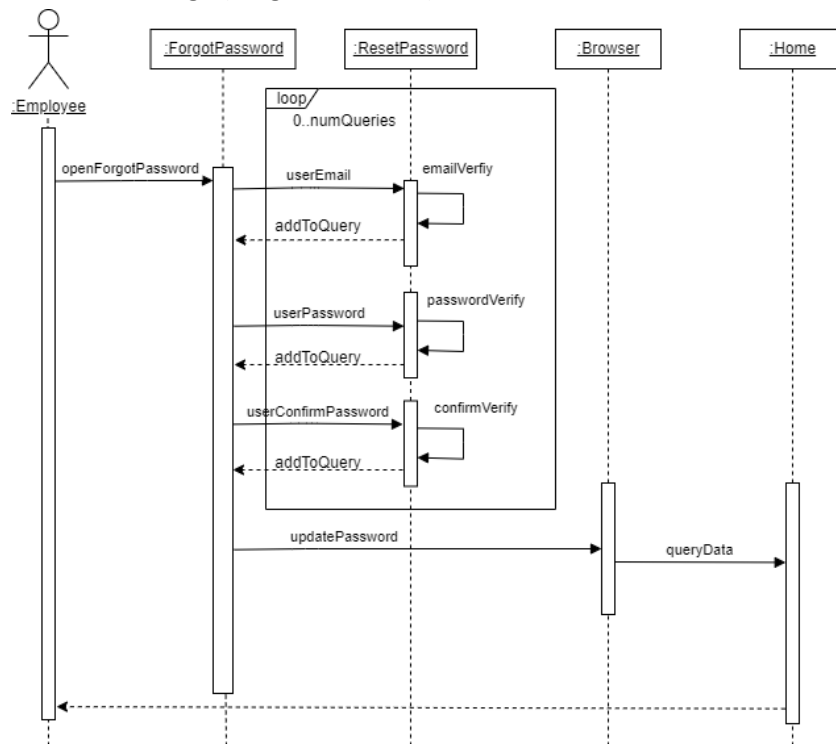
User Login:



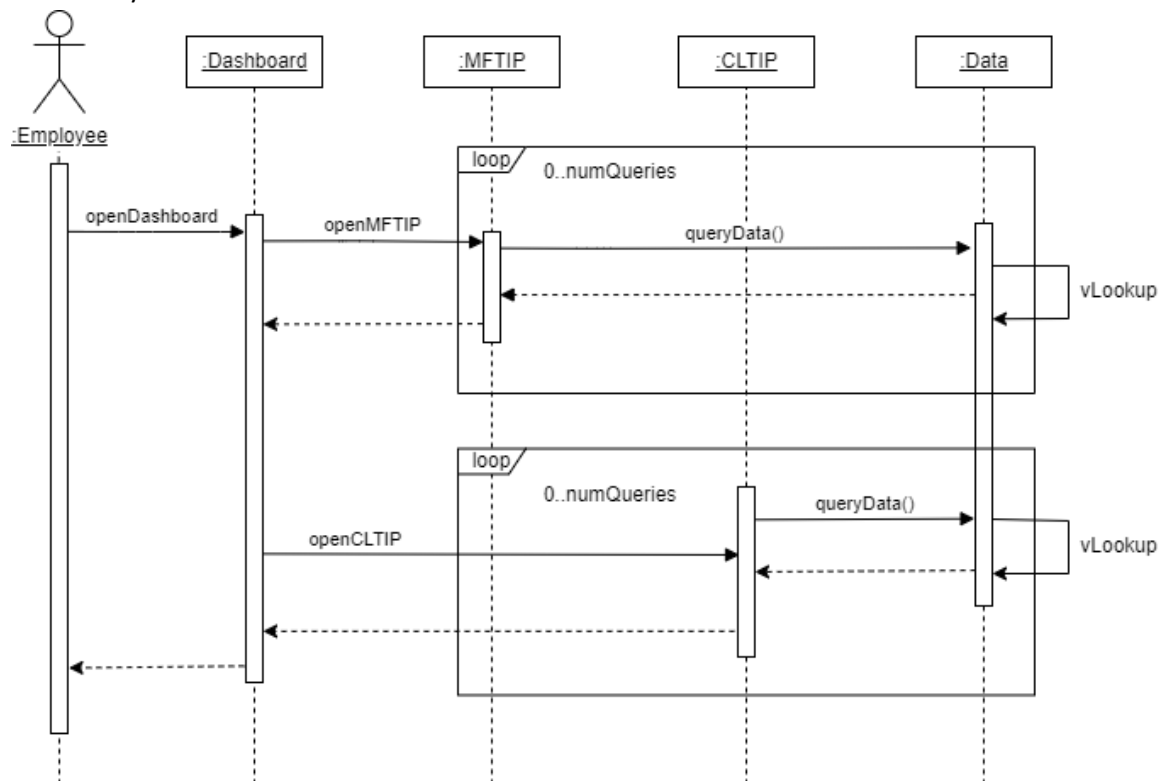
User Logout:



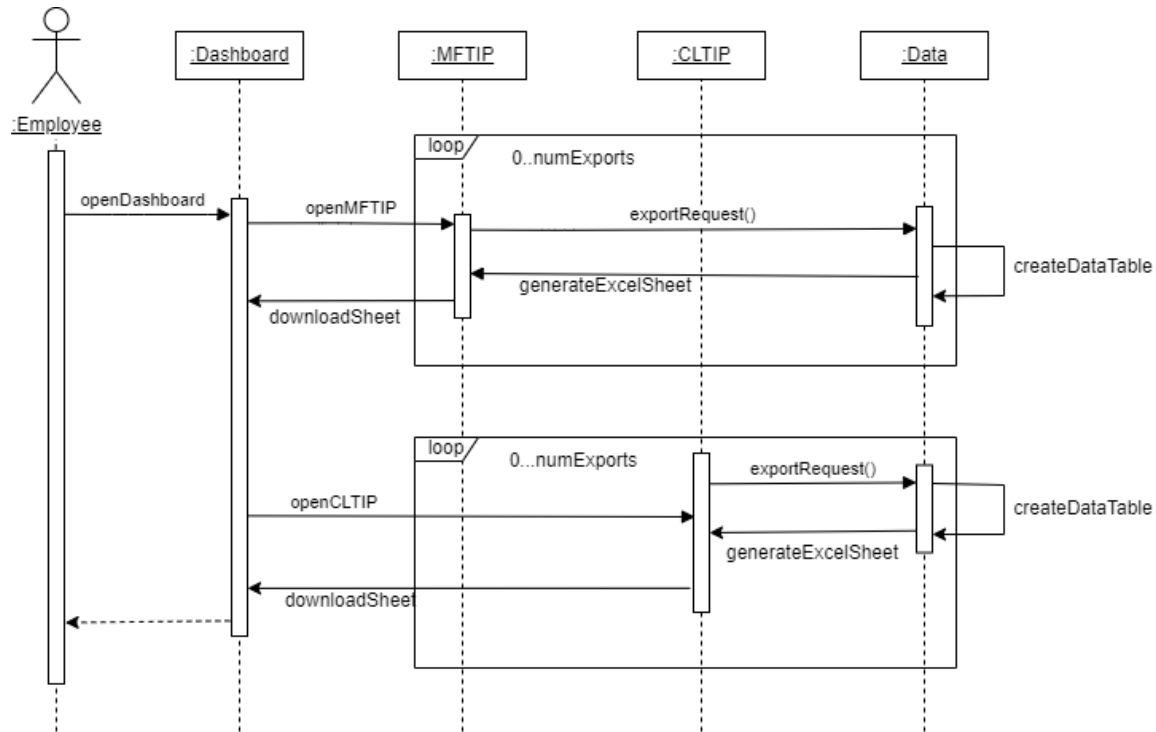
## Password Change (Forgot Password):



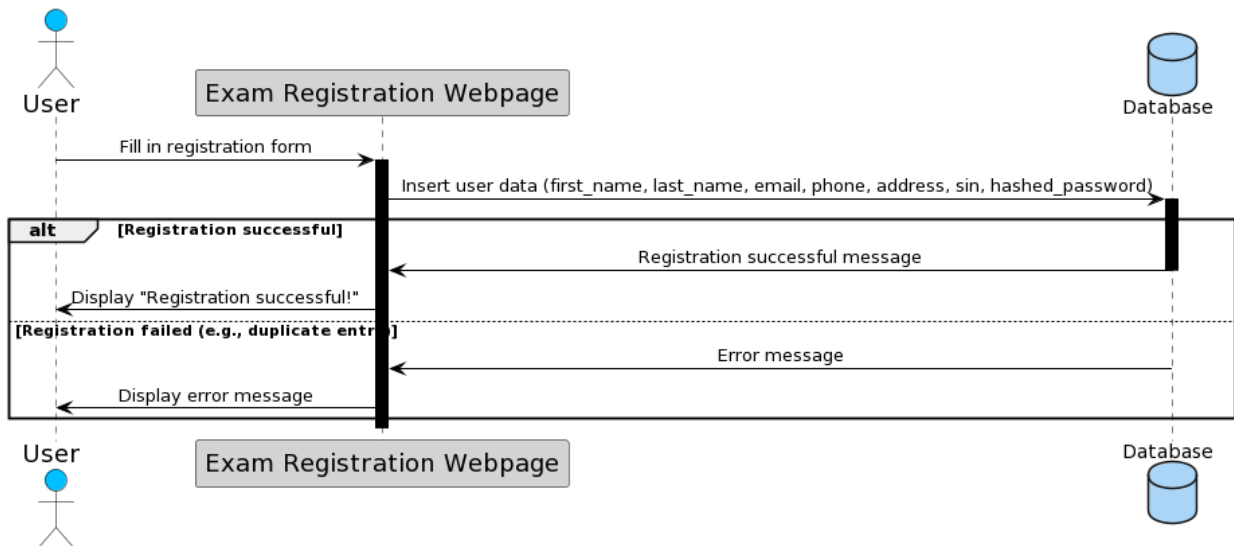
## Staff Query Search:



## Export Data:



## MFPA Exam Registration:



## Security Considerations

The website uses PHP's built-in `filter_var()` function with the `FILTER_SANITIZE_STRING` flag to sanitize user inputs before using them in database queries. This helps prevent potential security risks such as SQL injection attacks.

When looking into our implementation, several security concerns came up regarding account accessibility and user functionality of public services by different user groups, as such we divided the data made available based on the user authorities. The use of three unique user roles were taken into consideration: MPFAs, the public landowners and the MNRF staff. Since our client's current requirements stated that both MFTIP and CLITP programs shared the same internal portal and client data, a combination of services from both of the programs was made available to the generalized MRNF staff account

Several other security concerns such as data loss, data hijacking which are out of our scope, are related to the database hosting site itself as for this prototype project, Loki and phpMyAdmin do not have the capability to provide upscaling once realtime data gets imported into the database schema.

## Deployment and Hosting

Users will need loki access with the correct credentials to access the prototype.

## Testing and Quality Assurance:

Testing document for the whole project has been attached separately.

## Future Enhancements and Roadmap

Due to the limitation of software, time and manpower in this large-scale project, several features from the development process were scoped down. These include the ability to upload new client data, modify existing account data, add new staff accounts, create personalized selection for queries on several of the data information tables.

If this project was to be implemented in the next available academic year of COIS 4000Y, the implementation of the features mentioned above would uplift the functionality of this project to meet with the functionality of the current testing environment at the MNRF.

Future Enhancements and Milestones A. Improved application status updates and notifications (Q3 2023)

1. Real-time application status updates
2. Email and SMS notifications for landowners

Enhanced MFP approver profiles (Q4 2023)

1. Integration of ratings and reviews
2. Enhanced search and filter options

#### Comprehensive MFP dashboard for approvers (Q1 2024)

1. Overview of active applications and approvals
2. Performance metrics and reporting
3. Efficient communication tools

#### Advanced analytics and reporting for the ministry (Q2 2024)

1. Comprehensive reports on MFTIP and CLTIP program performance
2. Data visualization tools for easier analysis
3. Customizable report generation

#### Mobile app development for landowners and MFP approvers (Q3 2024)

1. Mobile app for Android and iOS platforms
2. Seamless integration with the website
3. Push notifications for important updates

#### Integration with other Ministry of Natural Resources and Forestry systems (Q4 2024)

1. Data sharing and collaboration with other relevant systems
2. Streamlined user experience across platforms

#### User Feedback and Continuous Improvement

1. Establishing a user feedback mechanism
2. Periodic user surveys and interviews
3. Prioritizing feature enhancements based on user feedback
4. Regular updates and maintenance