Password Manager Application Report

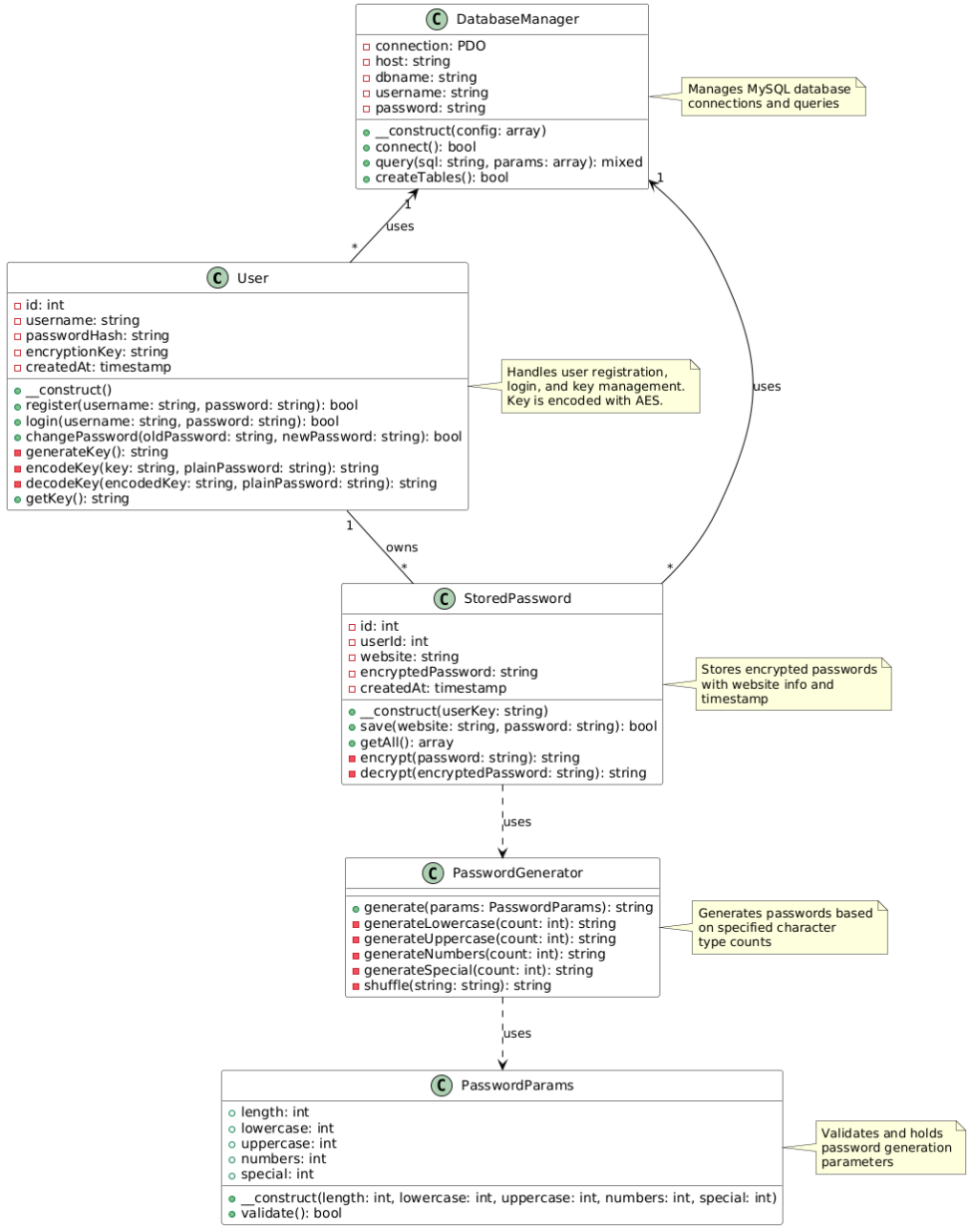
1. Project Overview

This PHP-based password manager application implements Object-Oriented Programming principles to provide secure password generation and storage functionality.

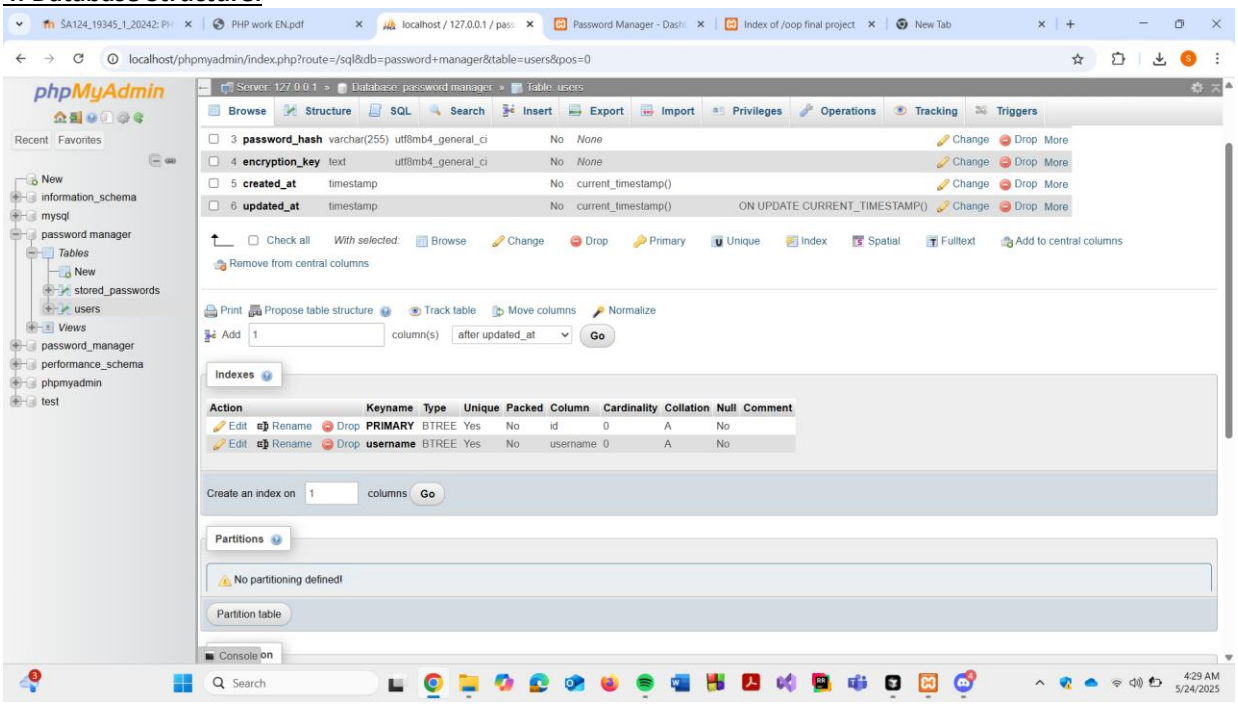
2. Features

- User registration and login system
- Secure password generation with customizable parameters
- AES-256 encryption for stored passwords
- User-specific encryption keys
- Modern and responsive user interface

3. UML Class Diagram:



4. Database Structure:



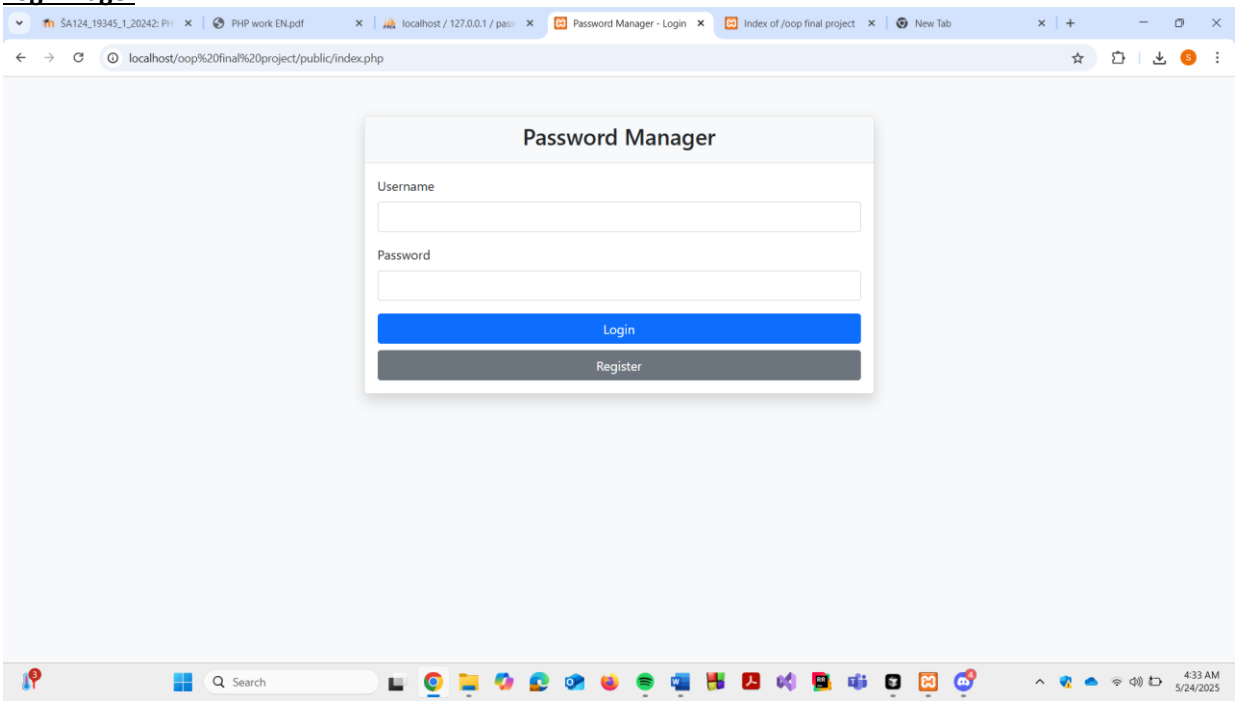
Tables:

1. users
   - id (Primary Key)
   - username
   - password_hash
   - encryption_key
   - created_at

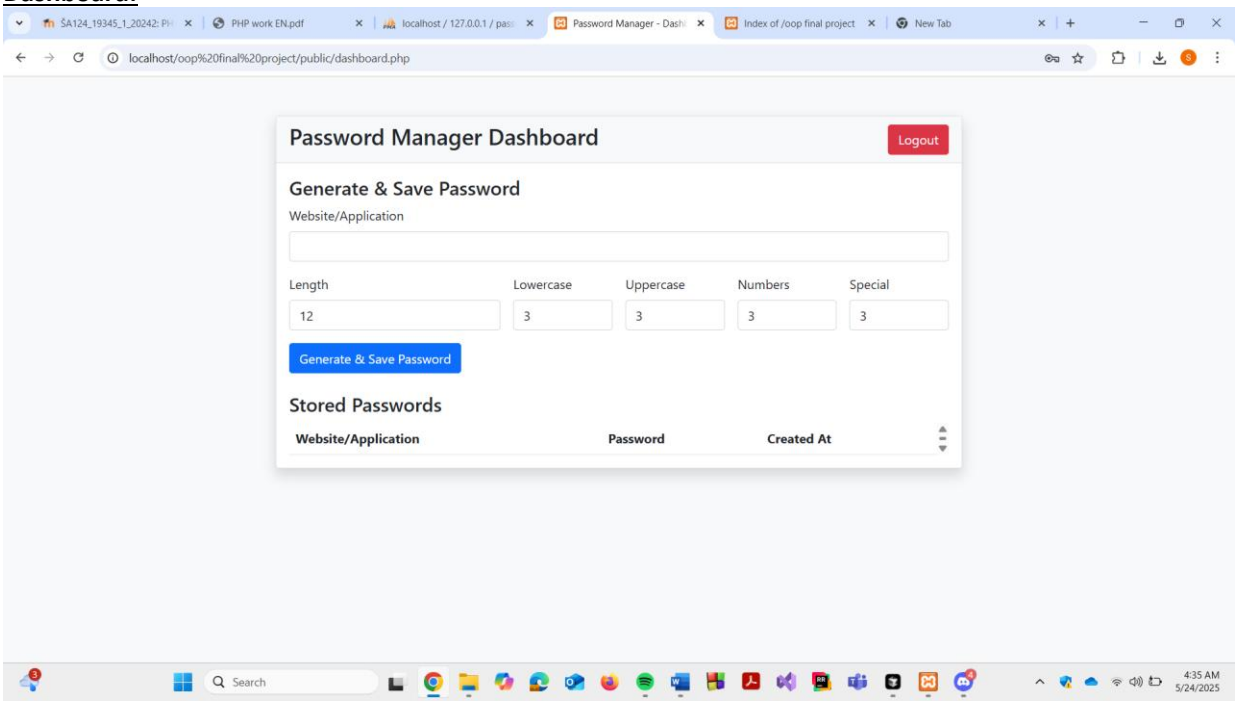
2. passwords
   - id (Primary Key)
   - user_id (Foreign Key)
   - website
   - password (encrypted)
   - created_at

5. Application Screenshots:

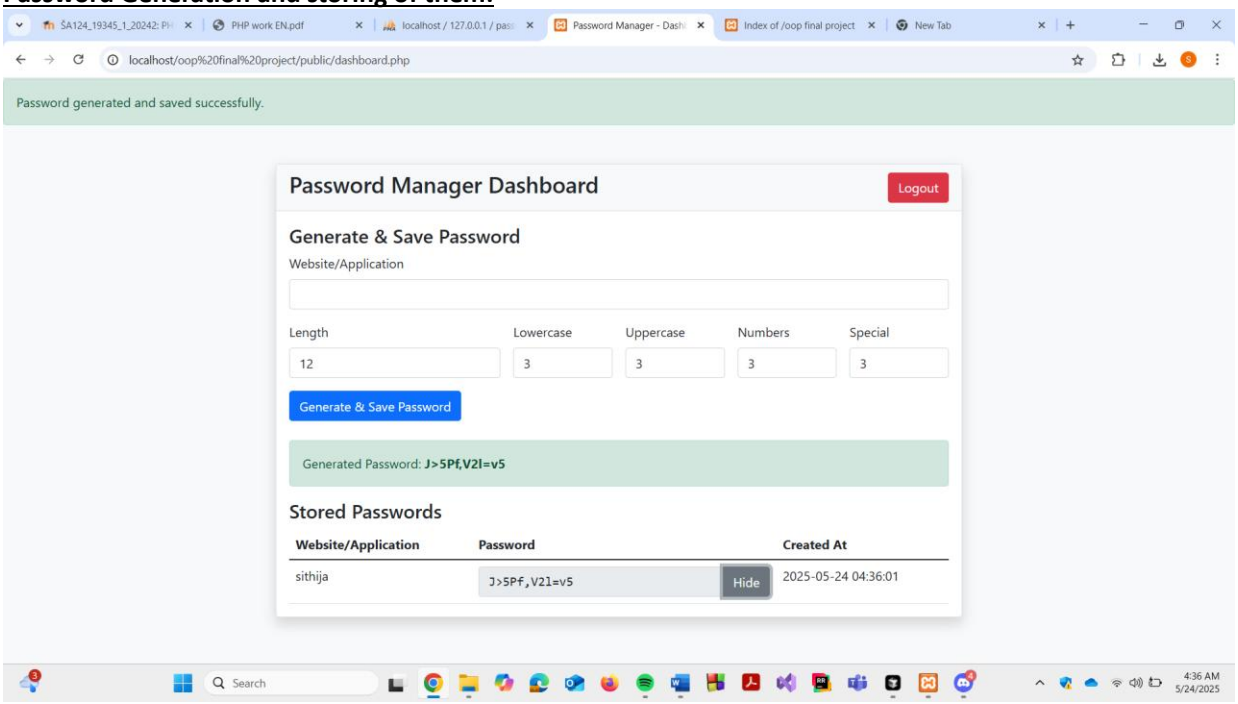
Login Page:



Dashboard:



Password Generation and storing of them:



6. Security Features:

1. Password Hashing
   - User passwords are hashed using bcrypt
   - Prevents plain-text password storage

2. Encryption
   - AES-256-CBC encryption for stored passwords
   - Unique encryption key per user
   - Key is encrypted with user's password

3. Session Security
   - Session-based authentication
   - Secure session handling
   - Automatic logout on session expiry

4. SQL Injection Prevention
   - Prepared statements
   - PDO for database operations

7. Code Structure:

project/
├— config/

```
|   └── database.php
├── src/
|   ├── autoload.php
|   └── classes/
|       ├── Database.php
|       ├── User.php
|       ├── Password.php
|       └── PasswordGenerator.php
└── public/
    ├── css/
    |   └── style.css
    ├── index.php
    ├── auth.php
    ├── dashboard.php
    └── logout.php
```

8. Testing Instructions:

1. Database Setup:
   - Create database 'password_manager'
   - Import database.sql

2. Application Setup:
   - Configure database connection in config/database.php
   - Place files in web server directory
   - Ensure proper permissions

3. Testing Steps:
   a. User Registration
   b. User Login
   c. Password Generation
   d. Password Storage
   e. Password Retrieval
   f. Logout

9. Requirements I made sure to make:

1. PHP OOP Implementation
2. MySQL Database Storage
3. User Authentication with Hashed Passwords
4. AES Encryption Implementation
5. Custom Password Generator
6. Secure Password Storage
7. UML and Database Documentation