PASSWORD MANAGER APP

Nurbaev Rasulbek, Kokumova Aidana, Rakhmatova Leyla

PROJECT OVERVIEW

This application is a simple yet functional password manager developed using JavaFX. It allows users to:

- Log in using a master password
- Store encrypted passwords for websites
- Generate strong passwords
- View and decrypt stored passwords
- Persist data in JSON format

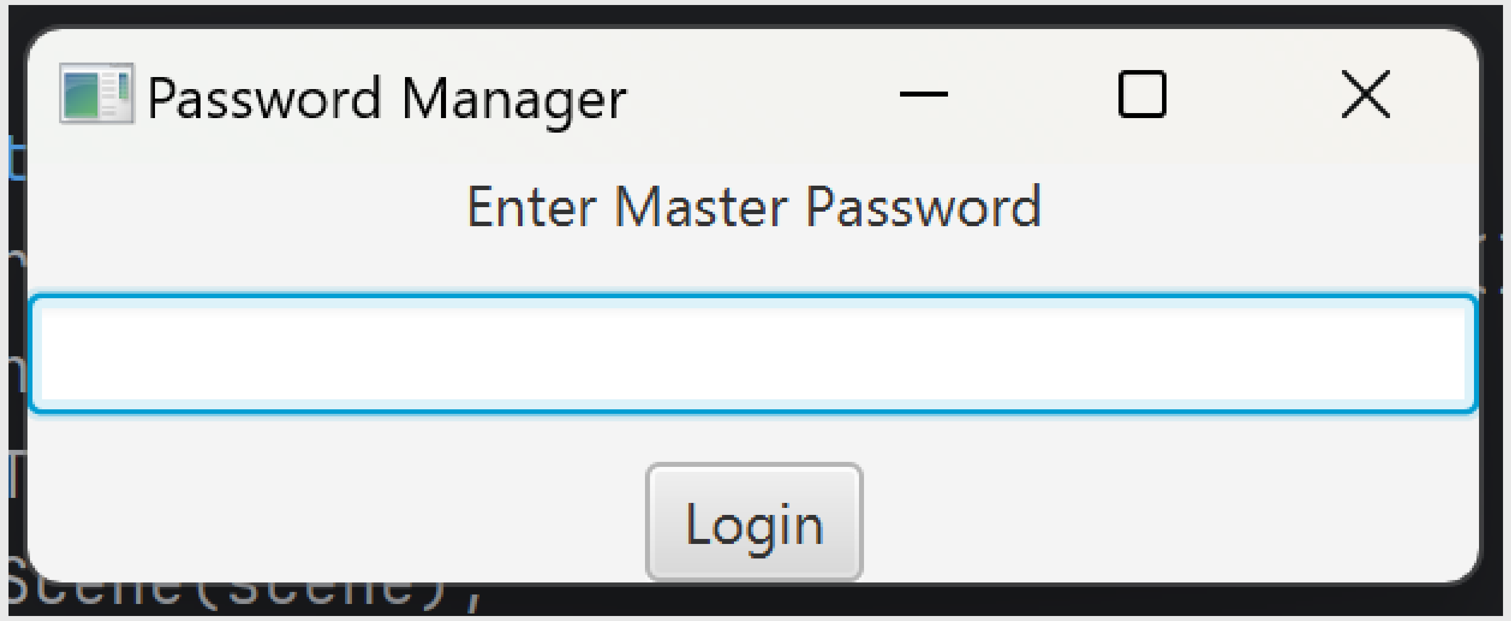
TECHNOLOGIES USED

- Java 17+
- JavaFX for GUI
- Gson for JSON serialization
- Base64 for encryption (replaceable with AES for stronger security)
- FXML for UI design

APPLICATION ARCHITECTURE

- Main.java – launches the app and loads login-view.fxml
- LoginController.java – handles master password authentication
- ManagerController.java – manages the main password logic
- PasswordEntry.java – data model for one password record
- PasswordUtil.java – password generation & (de/en)cryption
- StorageService.java – handles saving/loading JSON data

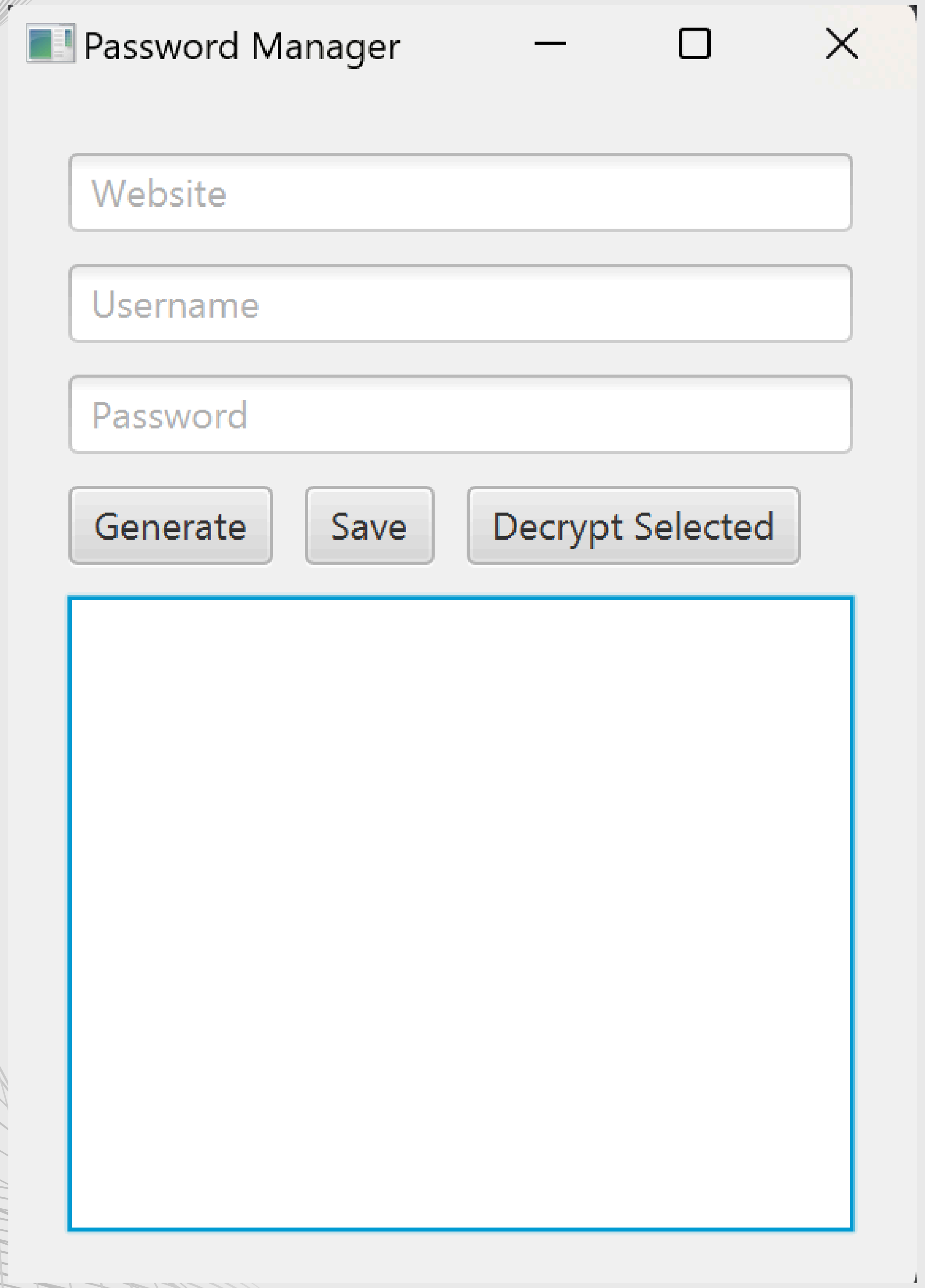
LOGIN INTERFACE



PasswordField → Input for master password

Login Button →Triggers onLoginClick()
→ Validates password
→ Loads manager-view.fxml if valid
→ Shows error if invalid

MAIN INTERFACE (PASSWORD MANAGER)



Website & Username Fields → Input for account info
Password Field → Input for password

Generate → Creates random password
Save → Encrypts and saves new entry
Decrypt Selected → Shows decrypted password in dialog
ListView → Shows stored entries (website - username)

PASSWORD ENCRYPTION LOGIC

In PasswordUtil.java:

```
public static String encrypt(String password) {
return Base64.getEncoder().encodeToString(password.getBytes());
 }
public static String decrypt(String encrypted) {
return new String(Base64.getDecoder().decode(encrypted));
 }
```

PASSWORD GENERATION

```
private static final String CHARACTERS =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789!@#$%"
;

public static String generatePassword(int length) {
    Random rand = new Random();
    StringBuilder sb = new StringBuilder(length);
    for (int i = 0; i < length; i++) {
        sb.append(CHARACTERS.charAt(rand.nextInt(CHARACTERS.length())));
    }
    return sb.toString();
}
```

Generates strong, random passwords using a mix of characters. Easily extendable to add rules.

JSON STORAGE

Passwords are saved in data/passwords.json

StorageService.java handles:

- savePasswords(List<PasswordEntry>)
- loadPasswords() → returns List<PasswordEntry>

Using Gson to serialize/deserialize the list of PasswordEntry.

PLANNED DATABASE INTEGRATION ARCHITECTURE

PostgreSQL
- Users Table
- Passwords Table

Benefits: Secure and scalable Enables multi-user support Better performance with indexing and queries Backup and restore support

Summary

Secure login with master password
Password encryption and generation
Clean JavaFX GUI
Modular, extensible code
PostgreSQL planned for reliable data storage