VILNIUS UNIVERSITY
ŠIAULIAI ACADEMY

BACHELOR PROGRAMME SOFTWARE ENGINEERING

Final Project

student: Yezholov Kyrylo

Lecturer: Asist. dr. Dervinis Donatas

Šiauliai, 2025

I. Introduction

Project Title: PHP OOP Password Management System.

Project Purpose: to create program page for PHP OOP passwords for generating and for storing in DB

GitHub URL: https://github.com/yezholov/PHP_FinalProject

II. System Architecture. UML and Database architecture

UML.

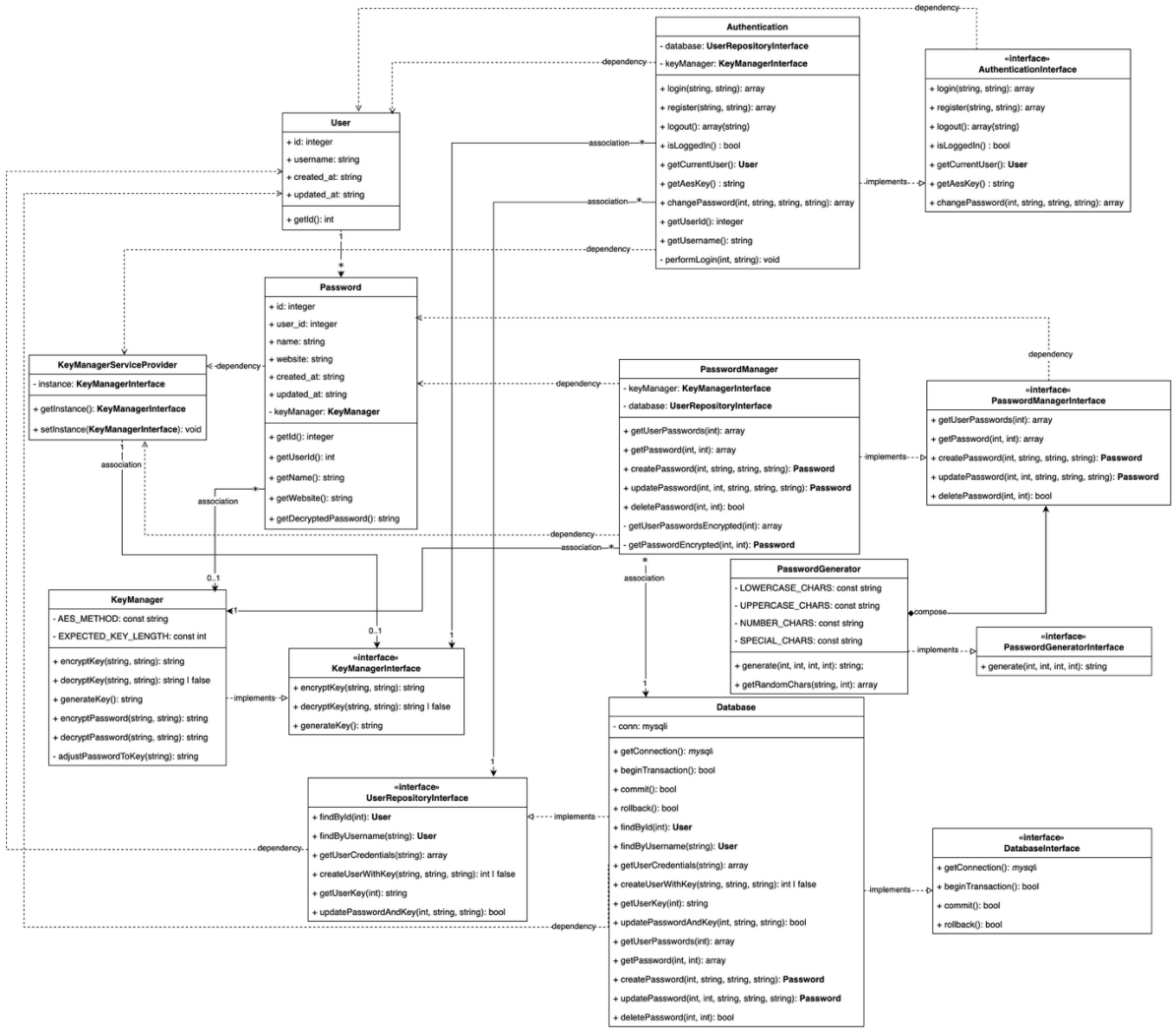

Fig. 1. UML Class Diagram

Database Schema.

The database dump (without data) is in the archive under the name "FinalProject_db.sql".

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id** 🔑	int			No	*None*		AUTO_INCREMENT
2	**username** 🔑	varchar(255)	*utf8mb4_unicode_ci*		No	*None*		
3	**password_hash**	varchar(255)	*utf8mb4_unicode_ci*		No	*None*		
4	**created_at**	timestamp			Yes	CURRENT_TIMESTAMP		DEFAULT_GENERATED
5	**updated_at**	timestamp		on update CURRENT_TIMESTAMP	Yes	CURRENT_TIMESTAMP		DEFAULT_GENERATED ON UPDATE CURRENT_TIMESTAMP

Fig. 2. Structure of the *users* table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**user_id**	int			No	*None*		
2	**aes_key_encrypted**	text	*utf8mb4_unicode_ci*		No	*None*		
3	**created_at**	timestamp			Yes	CURRENT_TIMESTAMP		DEFAULT_GENERATED

Fig. 3. Structure of the *user_keys* table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	int			No	*None*		AUTO_INCREMENT
2	**user_id**	int			No	*None*		
3	**name**	varchar(255)	*utf8mb4_unicode_ci*		No	*None*		
4	**password_encrypted**	text	*utf8mb4_unicode_ci*		No	*None*		
5	**website**	varchar(255)	*utf8mb4_unicode_ci*		Yes	*NULL*		
6	**created_at**	timestamp			Yes	CURRENT_TIMESTAMP		DEFAULT_GENERATED
7	**updated_at**	timestamp		on update CURRENT_TIMESTAMP	Yes	CURRENT_TIMESTAMP		DEFAULT_GENERATED ON UPDATE CURRENT_TIMESTAMP

Fig. 4. Structure of the *passwords* table.

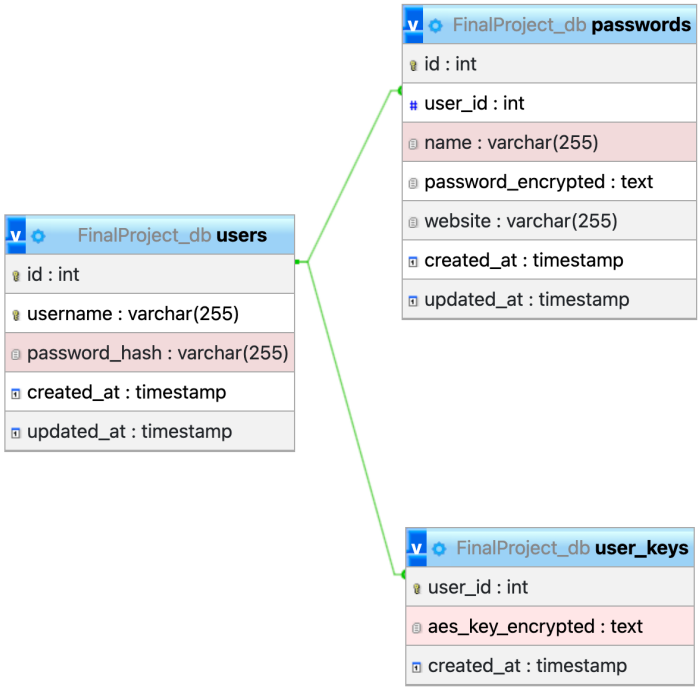

Fig. 5. *Designer view* in phpMyAdmin

III. Screenshots

1. **User registration**

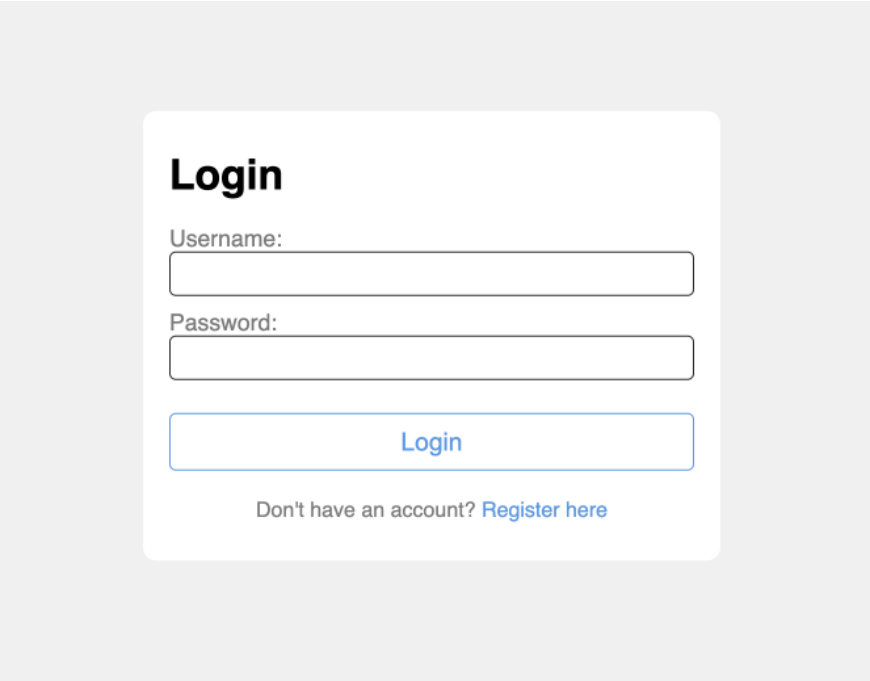

Fig. 6. Login Page

Since we don't have an account yet, we'll click on the "*Register here*" button to create one.

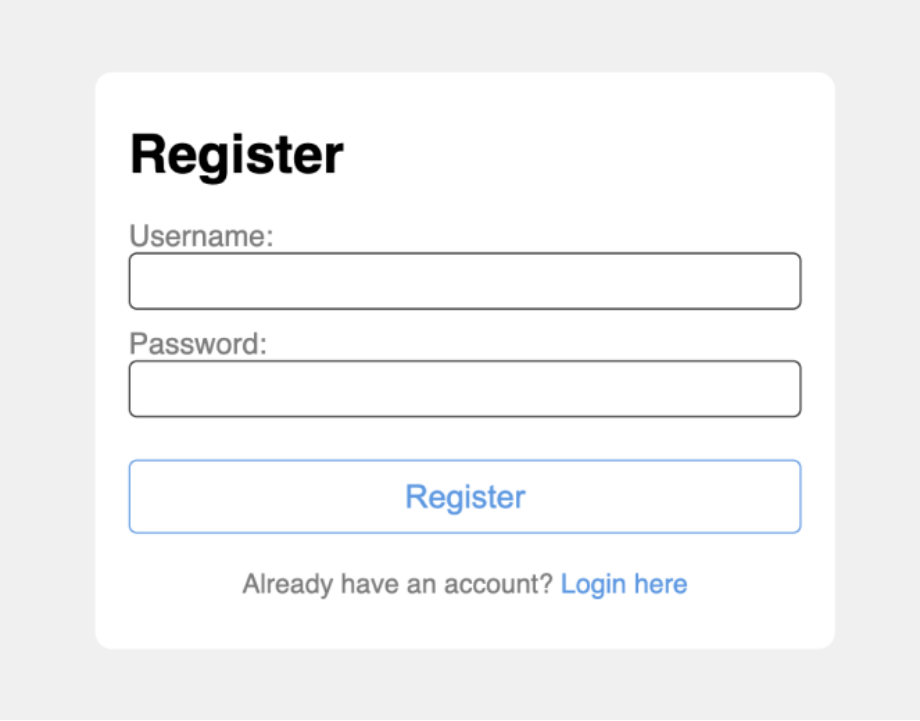

Fig. 7. Registration page

Let's go ahead and create a user with the name *user*.

Once the user is created, we're taken to the dashboard, where we can see the 'Change password' and 'Logout' buttons in the top-right corner.

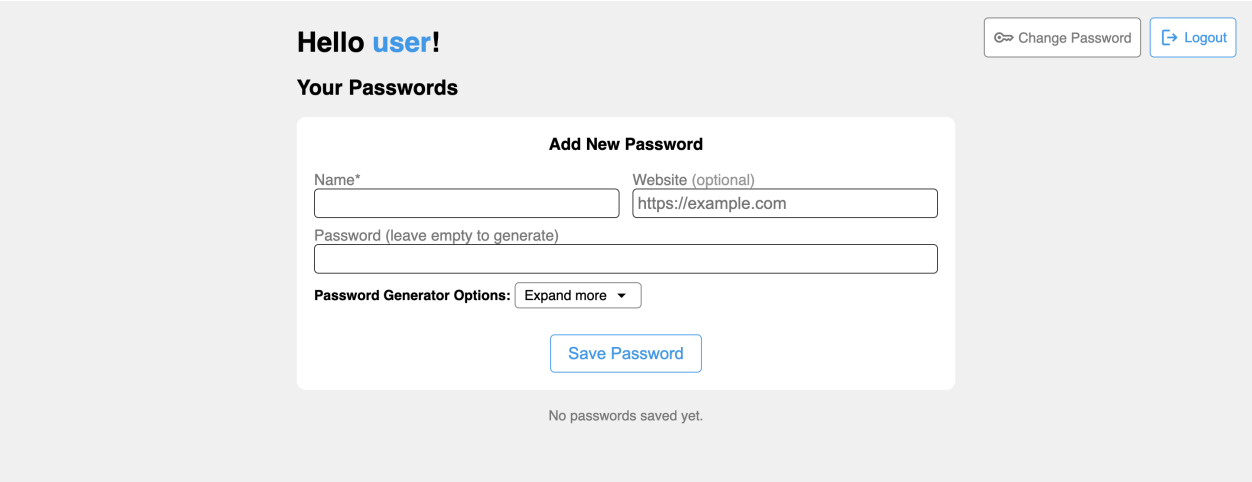

Fig. 8. Dashboard page.

Now, let's look at

```
SELECT * FROM `users`;
```

	id	username	password_hash	created_at	updated_at
Edit Copy Delete	6	user	$2y$10$eZ9vHQEwTQgj9Fh1pl5zzu2u7l/pEBxx0cCOBee.7Cr...	2025-05-10 15:55:09	2025-05-10 16:12:37

```
SELECT * FROM `user_keys`;
```

	user_id	aes_key_encrypted	created_at
Edit Copy Delete	6	KrwH4M738UFj6p6B2oR1MuFCfQEHS3c9Q+jRwza2NZA=	2025-05-10 15:55:09

database:

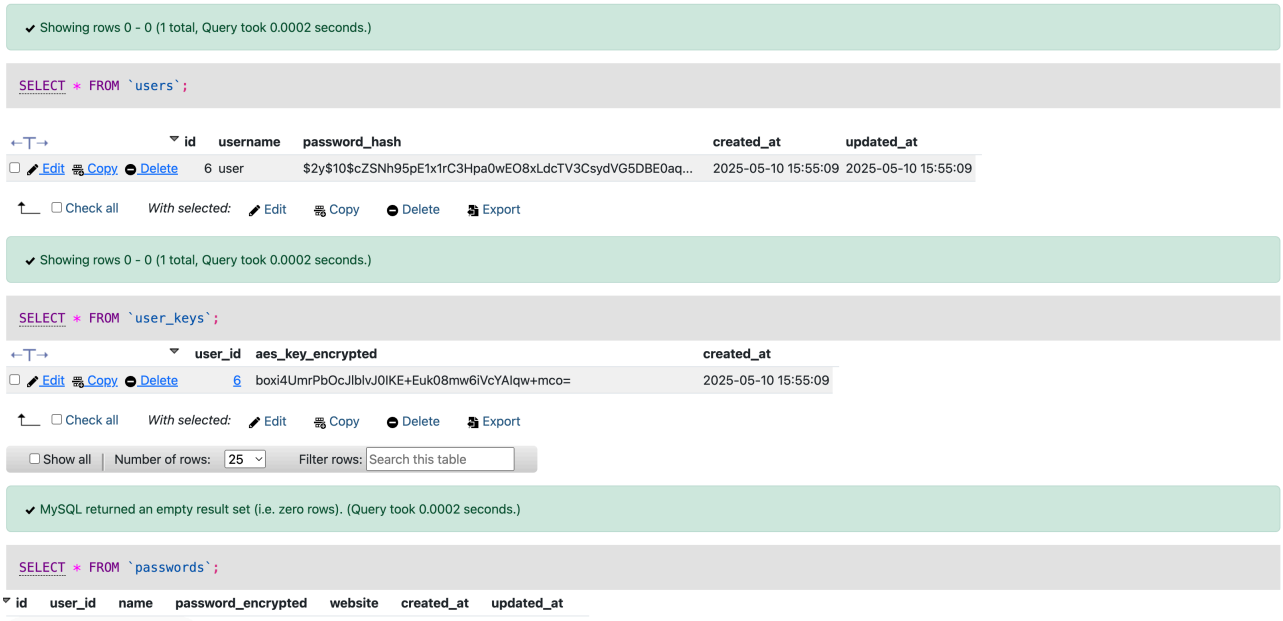

Fig. 9. Database after creation a user.

2. User login

After the user is created, we can try to log in again, but we will try to login with incorrect password.

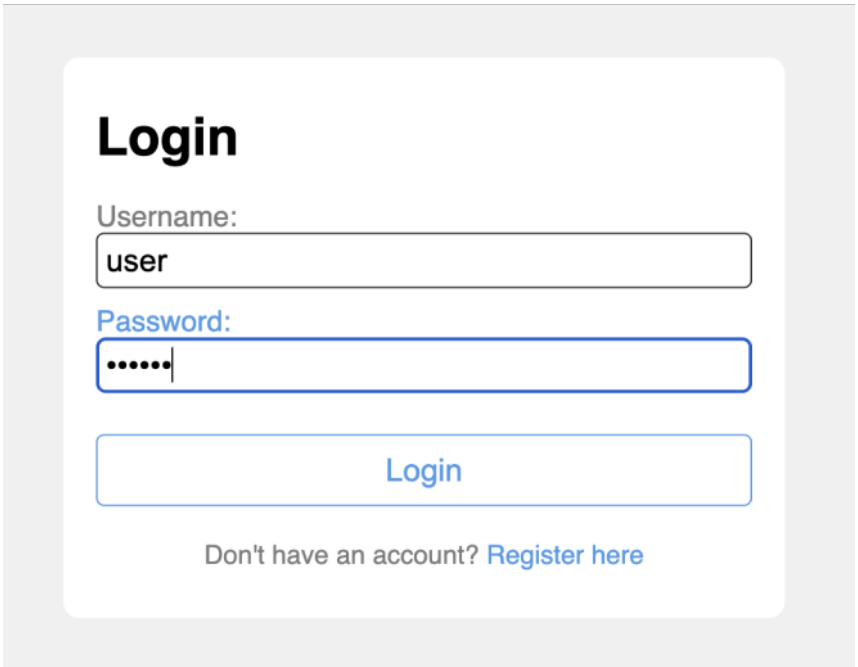

Fig. 10. Login with '*user*' username with incorrect password

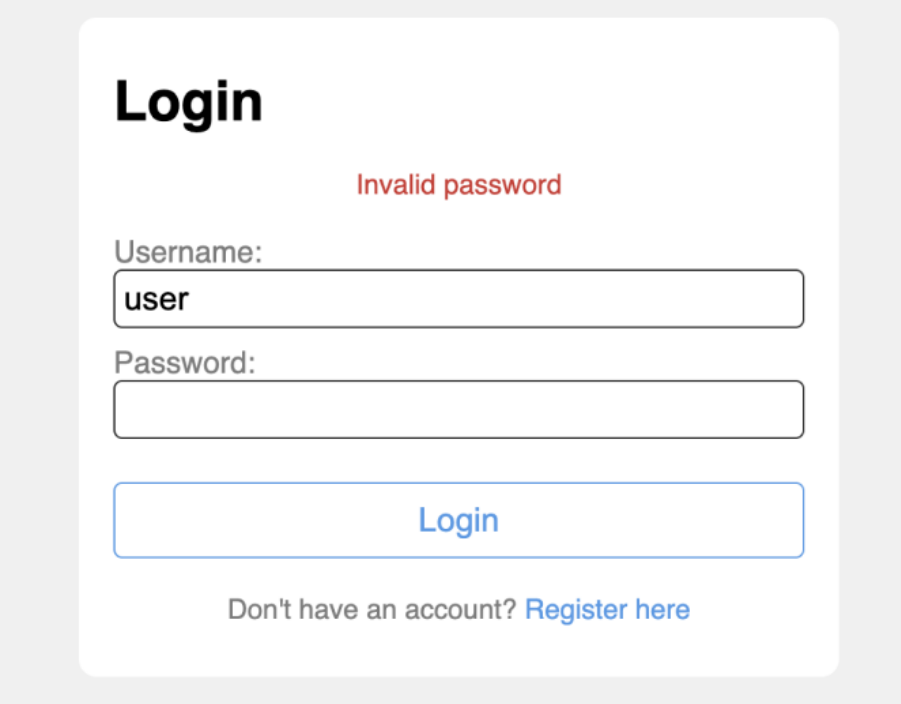

Fig. 11. Login with incorrect password.

We got error, so now we will try to login with correct password, and get Dashboard page(fig. 8)

3. Change password.

Look at fig. 8. We will try to change our user's password, let's click on "Change password".

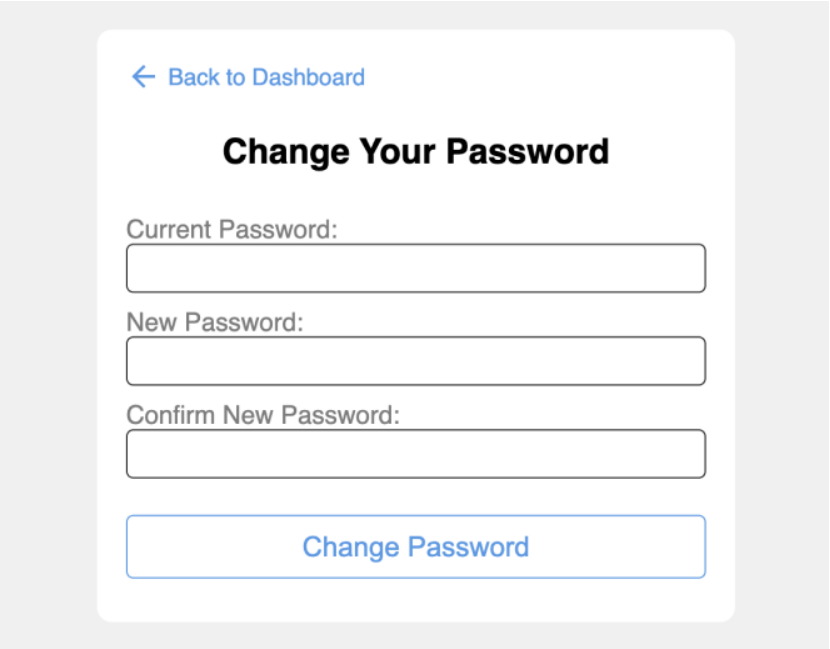

Fig. 12. Change password page

Let's try to enter incorrect current password.

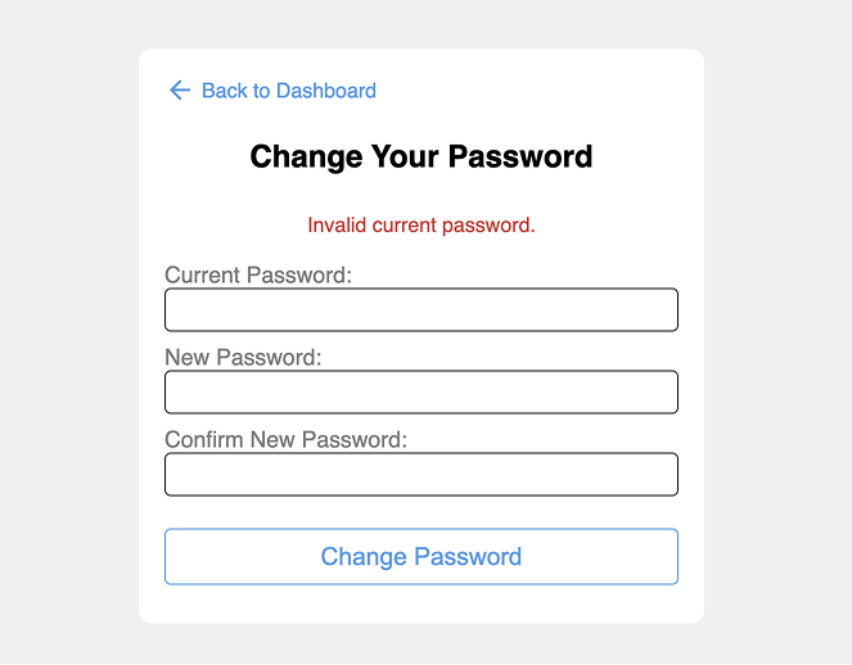

Fig. 13. Change password page. Wrong current password

Now let's enter right current password, but not the same new password.

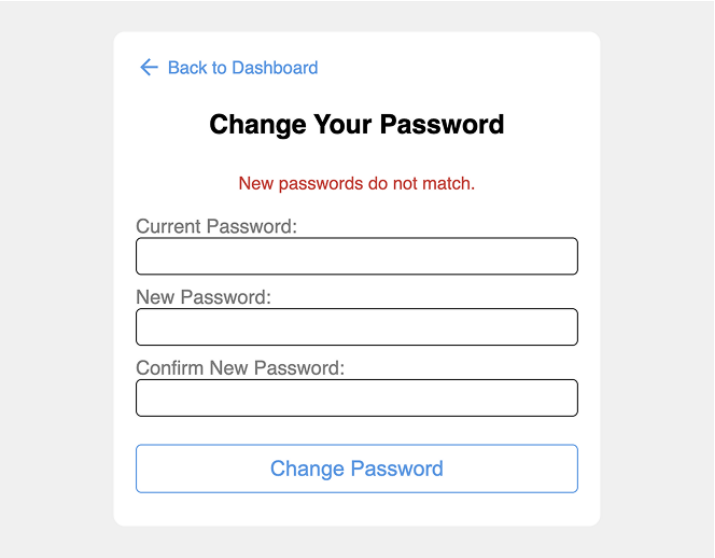

Fig. 14. Change password page. New passwords do not match.

Now we will fill it in correctly and try to change the password.

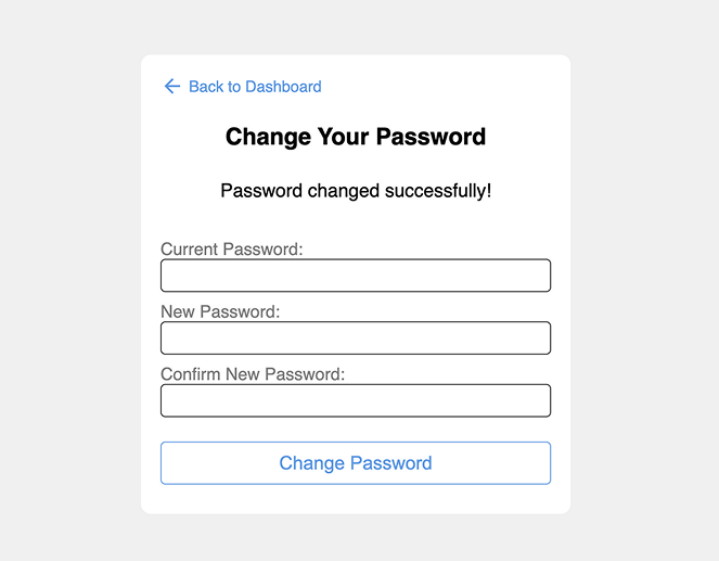

Fig. 15. Change password page. Success

Now, let's look at database:

```
SELECT * FROM `users`;
```

←T→	id	username	password_hash	created_at	updated_at
Edit Copy Delete	6	user	$2y$10$eZ9vHQEwTQgj9Fh1pl5zzu2u7l/pEBxx0cCOBee.7Cr...	2025-05-10 15:55:09	2025-05-10 16:12:37

```
SELECT * FROM `user_keys`;
```

←T→	user_id	aes_key_encrypted	created_at
Edit Copy Delete	6	KrwH4M738UFj6p6B2oR1MuFCfQEHS3c9Q+jRwza2NZA=	2025-05-10 15:55:09

Fig. 16. Database after changing password.

If we compare with fig. 9, the password hash has changed, and a new date has been written in *updated*_at. The encrypted *aes_key* has also changed.

4. Add password into password manager

Let's back to dashboard and try to create a new password into password manager.

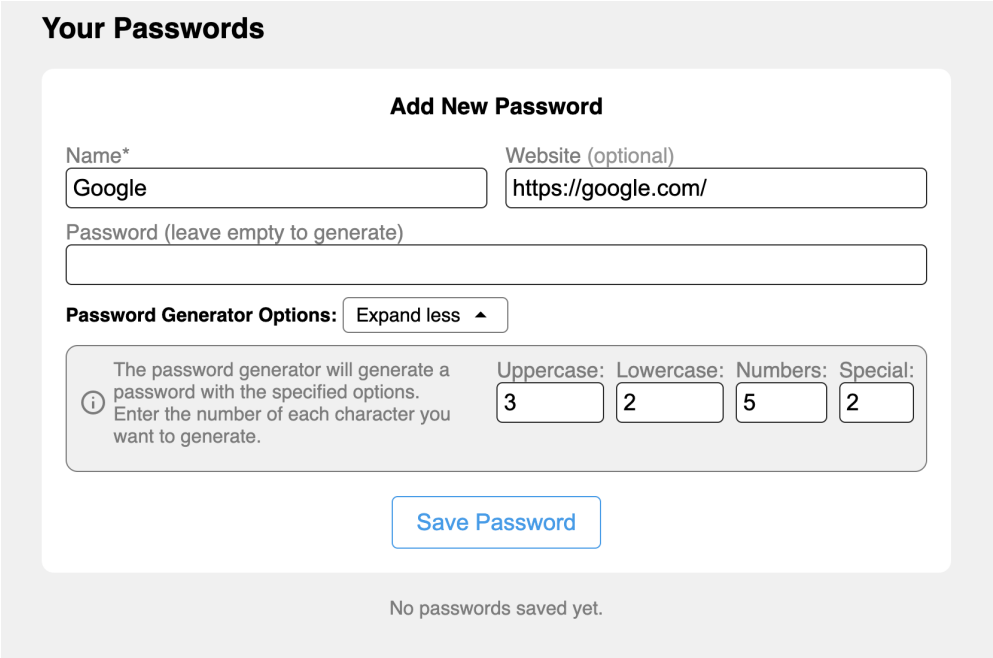

Fig. 17. Create a new password with auto-generated password

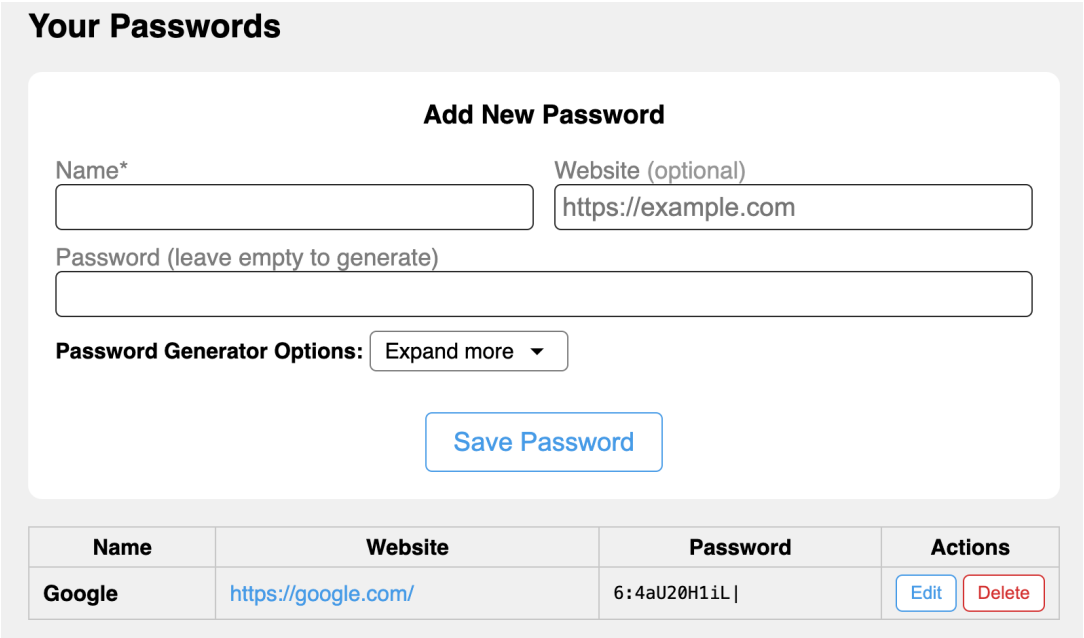

Fig. 18. Result of creation a new password with auto-generated password

```
SELECT * FROM `passwords`;
```

id	user_id	name	password_encrypted	website	created_at	updated_at
12	6	Google	sw/XVLnvDUa8jMpDlRfC8Q==	https://google.com/	2025-05-10 16:21:05	2025-05-10 16:21:05

Fig. 19. New password into Database, with encrypted password

Now let’s create a new password without website, and with manual password entry.

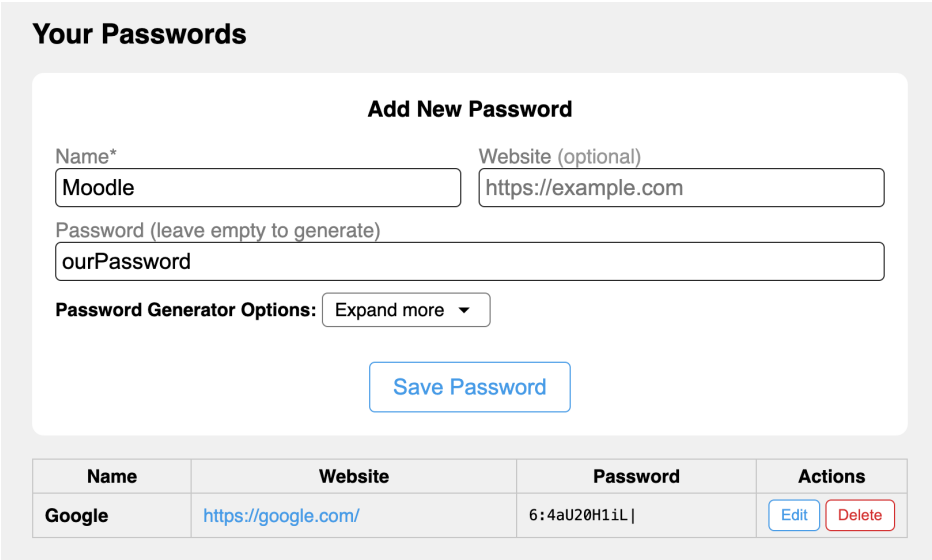

Fig. 20. Create a new password with manual password

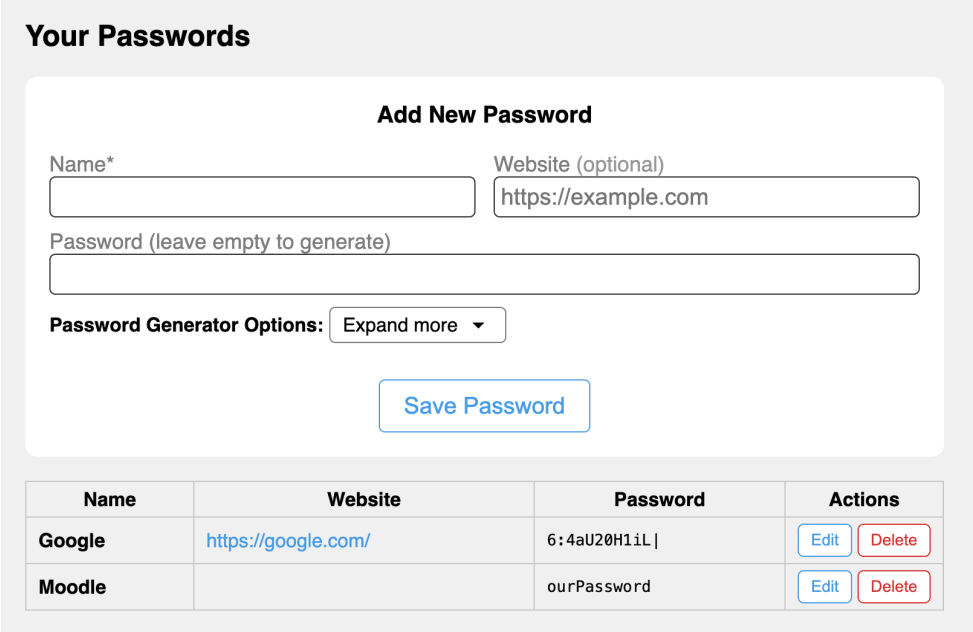

Fig. 21. Result of creation a new password with manual password

```
SELECT * FROM `passwords`;
```

id	user_id	name	password_encrypted	website	created_at	updated_at
12	6	Google	sw/XVLnvDUa8jMpDlRfC8Q==	https://google.com/	2025-05-10 16:21:05	2025-05-10 16:21:05
13	6	Moodle	sPboAulfwYS1kZObLJdCuw==		2025-05-10 16:24:51	2025-05-10 16:24:51

Fig. 22. Password into Database, with encrypted password.

Change the password generation parameters, for example: 5 uppercase, 2 lowercase, 2 numbers and one special character.

Name	Website	Password	Actions
Google	https://google.com/	`6:4aU20H1iL|`	Edit Delete
Moodle		`ourPassword`	Edit Delete
X (Twitter)	https://x.com	`ITR5a8D?Wf`	Edit Delete

Fig. 23. New password (https://x.com)

As you can see, the new password fully corresponds to our parameters.

Let's change the password, according to the 2nd point, and see if the passwords will be changed, as well as their appearance in the database.

Name	Website	Password	Actions
Google	https://google.com/	`6:4aU20H1iL|`	Edit Delete
Moodle		`ourPassword`	Edit Delete
X (Twitter)	https://x.com	`ITR5a8D?Wf`	Edit Delete

Fig. 24. Dashboard after password change

There are no visible changes, let's see what's in the database.

✔ Showing rows 0 - 0 (1 total, Query took 0.0003 seconds.)

```
SELECT * FROM `users`;
```

id	username	password_hash	created_at	updated_at
6	user	$2y$10$g8avmCdmKvE5.vX3Qr6Sh.gWBn7yrvLMVwqvonMTbFX...	2025-05-10 15:55:09	2025-05-10 16:33:57

✔ Showing rows 0 - 2 (3 total, Query took 0.0005 seconds.)

```
SELECT * FROM `passwords`;
```

id	user_id	name	password_encrypted	website	created_at	updated_at
12	6	Google	sw/XVLnvDUa8jMpDlRfC8Q==	https://google.com/	2025-05-10 16:21:05	2025-05-10 16:21:05
13	6	Moodle	sPboAulfwYS1kZObLJdCuw==		2025-05-10 16:24:51	2025-05-10 16:24:51
14	6	X (Twitter)	6R1z0GTCJpydvMFVleeI0w==	https://x.com	2025-05-10 16:28:18	2025-05-10 16:28:18

✔ Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)

```
SELECT * FROM `user_keys`;
```

user_id	aes_key_encrypted	created_at
6	boxi4UmrPbOcJIblvJ0IKE+Euk08mw6iVcYAIqw+mco=	2025-05-10 15:55:09

Fig. 25. Database after changing password.

When comparing fig. 16, password hash, and encrypted AES key have changed. This means that all our data is securely encrypted and successfully decrypted after changing the password.

5. Update password in password manager

Let's change the name and site for our first password (Google).

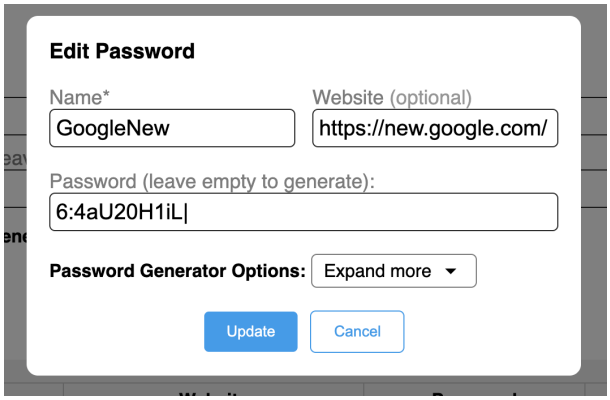

Fig. 26. Changing the name and website of one of the passwords.

Name	Website	Password	Actions
GoogleNew	https://new.google.com/	6:4aU20H1iL|	Edit Delete
Moodle		ourPassword	Edit Delete
X (Twitter)	https://x.com	ITR5a8D?Wf	Edit Delete

Fig. 27. New name and website for first password (ex. Google)

Now let's try to change the password for this same password to our own.

Name	Website	Password	Actions
GoogleNew	https://new.google.com/	newgooglepassword	Edit Delete
Moodle		ourPassword	Edit Delete
X (Twitter)	https://x.com	ITR5a8D?Wf	Edit Delete

Fig. 28. Manual password for first password

Now let's try to change the password to auto-generation for the same password.

Fig. 29. Update password with auto-generator

Fig. 30. New auto-generated password (GoogleNew)

As we can see, everything turned out successfully and our passwords are displayed.

6. Delete password in password manager

Now we will try to delete the first password.

Fig. 31. Password Manager after clicked on Delete on first password (GoogleNew)

We have successfully removed the password

IV. Conclusion

The developed PHP OOP Password Management System effectively addresses the need for secure password generation and storage. Through careful implementation of user authentication, AES key encryption, and intuitive password management features, the project successfully achieves its intended goals.