

ДОКУМЕНТАЦИЈА ЗА ЛАБОРАТОРИСКА ВЕЖБА З ПО ПРЕДМЕТОТ ИНФОРМАЦИСКА БЕЗБЕДНОСТ

1. Вовед

1.1 Цел на лабораториската вежба

Цел на оваа лабораториска вежба е имплементација на систем за авторизација и контрола на пристап базирана на улоги (Role-Based Access Control – RBAC), надграден врз системот за автентикација од претходната лабораториска вежба. Проектот содржи:

- Хиерархиски организациски улоги (ORG_ADMIN, ORG_USER, ORG_GUEST)
- Ресурсно-специфични улоги (USER_READER, USER_WRITER) според хиерархија со привремен пристап (just-in-time access)
- permission-based контрола на пристап

Оваа лабораториска вежба е изработена од Мартина Петковска со индекс број 223313.

Целиот документ може да се најде на GitHub:

https://github.com/mpetkovska27/IB-projects/tree/main/ib_lab3

1.2 Користени технологии

- Backend: PHP
- Frontend: HTML5, CSS
- База на податоци: SQLite

СТРУКТУРА НА ПРОЕКТОТ

1. **assets**
 - 1.1. **css**
 - 1.1.1. *style.css* – содржи стилови за изгледот на апликацијата
2. **database**
 - 2.1. *db_connection.php* – воспоставување конекција со SQLite база
 - 2.2. *db.sqlite* – SQLite база на податоци
 - 2.3. **migrations**
 - 2.3.1. *create_users_table.php* – креирање на табелата за корисници
 - 2.3.2. *create_roles_table.php* – креирање на табели за RBAC (roles, permissions)
 - 2.3.3. *admin_user.php* – иницијално креирање на admin корисник
3. **handlers**
 - 3.1. **auth**
 - 3.1.1. *login_handler.php* – обработка на логика за најава
 - 3.1.2. *register_handler.php* – обработка на регистрација на корисници
 - 3.1.3. *verify_2fa_handler.php* – верификација на двофакторска автентикација
 - 3.1.4. *logout_handler.php* – логика за одјава
 - 3.2. *save_handler.php* – зачувување и ажурирање на кориснички податоци
 - 3.3. *delete_handler.php* – бришење на корисник
 - 3.4. *request_jit_handler.php* – барање за Just-In-Time (JIT) пристап

4. **helpers**
 - 4.1. *session_helper.php* – управување со сесии и cookies
 - 4.2. *user_helper.php* – помошни функции поврзани со корисници
 - 4.3. *authorization_helper.php* – функции за контрола на пристап (RBAC)
 - 4.4. *two_factor_helper.php* – генерирање и испраќање 2FA кодови
 - 4.5. *password_validator.php* – валидација на лозинки
 - 4.6. *verify_email_with_code.php* – верификација на е-маил адреса со код
5. **pages**
 - 5.1. **auth**
 - 5.1.1. *login.php* – страница за најава
 - 5.1.2. *register.php* – страница за регистрација
 - 5.2. *form.php* – форма за креирање и ажурирање на корисници
6. **index.php**
 - 6.1. Главна почетна страница на апликацијата

2. ФУНКЦИОНАЛНОСТИ

Системот за авторизација и контрола на пристап имплементира повеќе функционалности кои овозможуваат безбедно управување со корисници, улоги и дозволи. Функционалностите се поделени според нивната намена и улога во системот.

1. Авторизација

- Системот обезбедува корисничка авторизација и контрола на пристап до апликацијата.
- Пристапот до различни делови на апликацијата се одобрува или одбива врз основа на улогите на корисникот.

2. Дефинирање и менаџирање на улоги

- Дефинирани се **организациски улоги** (organization-level roles) кои обезбедуваат широк пристап за целиот систем. ([ORG_ADMIN](#), [ORG_USER](#), [ORG_GUEST](#))
- Дефинирани се и **специфични за ресурс улоги** (resource-specific roles) кои обезбедуваат прецизни дозволи за конкретни ресурси, во согласност со принципот **least privilege**. ([USER_READER](#), [USER_WRITER](#))

3. Контрола на пристап (Role-Based Access Control, RBAC)

- Пристапот до функционалности се контролира врз основа на улоги и дозволи.
- Корисникот има пристап само ако улогата што ја поседува содржи соодветна дозвола.
- При регистрација на нов корисник автоматски се доделува ORG_USER улогата.
- Ненајавен корисник има улога ORG_GUEST.

4. Пермисии и дозвола за пристап

Дефинирани се 5 permissions:

view_public	Приказ на јавни секции
view_dashboard	Пристап до dashboard
view_account	Приказ на сопствени информации
view_all_users	Приказ на листа на корисници
manage_users	Креирање/ажурирање/бришење корисници

Пермисиите се мапираат со улогите преку табелата role_permissions:

- ORG_GUEST
 - view_public
- ORG_USER
 - view_public
 - view_dashboard
 - view_account
- ORG_ADMIN
 - view_public
 - view_dashboard
 - view_account
 - view_all_users
 - manage_users
- USER_READER
 - view_all_users
- USER_WRITER
 - manage_users

Улогите се доделуваат на корисници преку табелата user_roles:

- При регистрација: се доделува ORG_USER (трајна)
- Admin корисник: се доделува ORG_ADMIN (трајна)
- JIT пристап: може да се додели USER_READER или USER_WRITER (привремено, 10 секунди)
 - JIT улогите автоматски се одземаат по истекот на дозволениот временски период, со што се зачувува принципот least privilege и се минимизира ризикот од злоупотреба.

3. База на податоци

Сите улоги во системот се дефинирани во create_roles_table.php.

4.1 Табела: users

```
CREATE TABLE IF NOT EXISTS users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    username TEXT NOT NULL UNIQUE,
    email TEXT UNIQUE NOT NULL,
    password_hash TEXT NOT NULL,
    first_name TEXT,
    last_name TEXT,
    email_verified INTEGER DEFAULT 0,
    email_verification_code TEXT,
    email_verification_expires DATETIME,
    two_factor_code TEXT,
    two_factor_code_expires DATETIME,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

4.2 Табела: roles

```
CREATE TABLE IF NOT EXISTS roles (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT NOT NULL UNIQUE,
    type TEXT NOT NULL CHECK(type IN ('organization', 'resource')),
    hierarchy_level INTEGER,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

Организациски улоги (хиерархиски):

- ORG_ADMIN (hierarchy_level: 1) - Највисока улога
- ORG_USER (hierarchy_level: 2) - Стандардна улога
- ORG_GUEST (hierarchy_level: 3) - Најниска улога

Ресурсно-специфични улоги (JIT):

- USER_READER (hierarchy_level: 99) - Читање на корисници
- USER_WRITER (hierarchy_level: 99) - Менаџирање на корисници

4.3 Табела: permissions

```
CREATE TABLE IF NOT EXISTS permissions (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT NOT NULL UNIQUE,
    description TEXT NOT NULL,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

Достапни permissions:

- view_public - Приказ на јавни секции
- view_dashboard - Пристап до dashboard
- view_account - Приказ на сопствени информации
- view_all_users - Приказ на листа на корисници
- manage_users - Креирање, ажурирање и бришење на корисници

4.4 Табела: role_permissions

```
CREATE TABLE IF NOT EXISTS role_permissions (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    role_id INTEGER NOT NULL,
    permission_id INTEGER NOT NULL,
    FOREIGN KEY (role_id) REFERENCES roles(id) ON DELETE CASCADE,
    FOREIGN KEY (permission_id) REFERENCES permissions(id) ON DELETE CASCADE,
    UNIQUE(role_id, permission_id)
);
```

Мапирање на permissions по улоги:

- ORG_GUEST: view_public
- ORG_USER: view_public, view_dashboard, view_account

- ORG_ADMIN: view_public, view_dashboard, view_account, view_all_users, manage_users
- USER_READER: view_all_users
- USER_WRITER: manage_users

4.5 Табела: user_roles

```
CREATE TABLE IF NOT EXISTS user_roles (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    user_id INTEGER NOT NULL,
    role_id INTEGER NOT NULL,
    expires_at DATETIME DEFAULT NULL,
    assigned_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY(role_id) REFERENCES roles(id) ON DELETE CASCADE,
    FOREIGN KEY(user_id) REFERENCES users(id) ON DELETE CASCADE
);
```

Забелешка: expires_at се користи за JIT улоги. Ако е NULL, улогата е трајна.

4. Helper функции

4.1 Session функции (helpers/session_helper.php)

generateSessionToken()

setSessionTokenCookie(\$token)

clearSessionTokenCookie()

cleanupExpiredRoles(\$userId = null)

- Брише истечени JIT улоги од базата
- Ако е наведен \$userId, ги чисти само за тој корисник
- Се повикува автоматски при проверки на пристап

isLoggedIn()

requireLogin()

- Пренасочува на login.php ако корисникот не е најавен

requireGuest()

- Пренасочува на index.php ако корисникот е најавен

logout()

4.2 User функции (helpers/user_helper.php)

getCurrentUser()

getAllUsers()

[4.3 Authorization функции \(helpers/authorization_helper.php\)](#)

getUserRoles(\$userId)

- Враќа сите активни улоги за корисник (вклучувајќи JIT)
- Автоматски ги чисти истечените улоги
- Враќа: array

getUserPermissions(\$userId)

- Враќа сите permissions за корисник преку неговите улоги
- Враќа: array

hasRole(\$userId, \$roleName)

- Проверува дали корисникот има одредена улога
- Враќа: boolean

hasPermission(\$userId, \$permissionName)

- Проверува дали корисникот има одредено permission
- Враќа: boolean

assignJITRole(\$userId, \$roleName, \$expiresInSeconds = 10)

- Доделува JIT улога на корисник
- Автоматски ја брише претходната JIT улога од ист тип
- Времетраење: по default 10 секунди
- Враќа: boolean

requirePermission(\$permissionName)

- Проверува дали корисникот е најавен и има одредено permission
- Пренасочува на login.php ако не е најавен
- Пренасочува на index.php ако нема permission

[4.4 Two-Factor функции \(helpers/two_factor_helper.php\)](#)

generateVerificationCode(\$length = 8)

sendVerificationCode(\$email, \$code, \$type = 'registration')

[4.5 Password функции \(helpers/password_validator.php\)](#)

validatePassword(\$password)

5. Handlers

5.1 Register Handler ([handlers/auth/register_handler.php](#))

- Автоматски доделува ORG_USER улога

5.2 Login Handler ([handlers/auth/login_handler.php](#))

5.3 Verify 2FA Handler ([handlers/auth/verify_2fa_handler.php](#))

5.4 Verify Email Handler ([helpers/verify_email_with_code.php](#))

5.5 Save Handler ([handlers/save_handler.php](#))

Функционалност:

- Креира или ажурира корисник (за корисници со manage_users permission)
- Валидира username и email
- Опционално ажурира лозинка
- При креирање, автоматски доделува ORG_USER улога
- Враќа: пренасочување на index.php

5.6 Delete Handler ([handlers/delete_handler.php](#))

Функционалност:

- Брише корисник од базата (за корисници со manage_users permission)
- Автоматско бришење на поврзаните улоги (CASCADE)
- Враќа: пренасочување на index.php

5.7 Request JIT Handler ([handlers/request_jit_handler.php](#))

Функционалност:

- Доделува JIT улога (USER_READER или USER_WRITER)
- Времетраење: 10 секунди(само за тестирање)
- Автоматско истекување и бришење
- Враќа: пренасочување на index.php

5.8 Logout Handler ([handlers/auth/logout_handler.php](#))

Тестирање

1. Поставување на базата

```
php database/migrations/create_users_table.php  
php database/migrations/create_roles_table.php  
php database/migrations/admin_user.php
```

2. Регистрација: При регистрација на нов корисник автоматски му се доделува улога ORG_USER

3. Тестирање на организациски улоги

ORG_GUEST: може да види само јавни секции

The screenshot shows the homepage of the "Role-Based Access Control and Authorization System". At the top, there are "Login" and "Register" buttons. Below them, the title "Role-Based Access Control and Authorization System" is displayed. A sub-header states: "Lab Exercise 3, made by Martina Petkovska - 223313, that implements a complete role-based access control (RBAC) system with hierarchical organization roles, resource-specific roles, and just-in-time (JIT) access management." A "About This Lab Exercise" section contains a detailed description of the system's features, including secure user authentication, role-based access control, resource-specific roles, JIT access management, permission-based access control, and secure user management. At the bottom, a copyright notice reads: "© 2025 User Authentication System - Laboratory Exercise 3".

ORG_USER: може да види јавни и само свои информации

The screenshot shows the homepage of the "Role-Based Access Control and Authorization System" for a logged-in user named "mpet". At the top, there are "Welcome, mpet!" and "Logout" buttons. Below them, the title "Role-Based Access Control and Authorization System" is displayed. A sub-header states: "Lab Exercise 3, made by Martina Petkovska - 223313, that implements a complete role-based access control (RBAC) system with hierarchical organization roles, resource-specific roles, and just-in-time (JIT) access management." A "About This Lab Exercise" section contains a detailed description of the system's features, including secure user authentication, role-based access control, resource-specific roles, JIT access management, permission-based access control, and secure user management. A "Your Account Information" section displays the user's details: Username: mpet, Email: mpet@test.com, and Registered on: November 11, 2025 at 4:48 PM. It also includes two buttons: "Request Reader Permission" and "Request Writer Permission". At the bottom, a copyright notice reads: "© 2025 User Authentication System - Laboratory Exercise 3".

ORG_ADMIN:

- може да види јавни и само свои информации
- може да види сите корисници
- може да креира/ажурира/брише корисници

Auth System

Welcome, admin! [Logout](#)

Role-Based Access Control and Authorization System

Lab Exercise 3, made by Martina Petkovska - 223313, that implements a complete role-based access control (RBAC) system with hierarchical organization roles, resource-specific roles, and just-in-time (JIT) access management.

About This Lab Exercise

This application demonstrates a comprehensive authorization system implemented as part of Laboratory Exercise 3. The system provides:

- secure user authentication with two-factor authentication (2FA) and email verification,
- role-based access control (RBAC) with hierarchical organization-level roles (ORG_ADMIN, ORG_USER, ORG_GUEST),
- resource-specific roles (USER_READER, USER_WRITER) with least privilege principle,
- just-in-time (JIT) access management with automatic expiration and revocation,
- permission-based access control for different sections of the application,
- and secure user management with CRUD operations protected by role permissions.

Your Account Information

Username: admin
Email: admin@admin.com
Registered on: December 14, 2025 at 3:31 PM

[Request Reader Permission](#) [Request Writer Permission](#)

Registered Users

[Create User](#)

ID	Username	Email	Registered	Actions
14	admin	admin@admin.com	Dec 14, 2025	Edit Delete
9	vbla	vbla@test.com	Nov 11, 2025	Edit Delete
7	ilat	ilat@test.com	Nov 11, 2025	Edit Delete
6	mpet	mpet@test.com	Nov 11, 2025	Edit Delete

Total users: 4

© 2025 User Authentication System - Laboratory Exercise 3

4. Тестирање на ресурсно-специфични улоги и JIT пристап

1. Најава како обичен корисник
2. Кликни "Request Reader Permission"
3. Провери: може да види листа на корисници
4. Почекај 10 секунди
5. Провери: пристапот е отстранет

USER_READER

Auth System

Welcome, mpet! [Logout](#)

Role-Based Access Control and Authorization System

Lab Exercise 3, made by Martina Petkovska - 223313, that implements a complete role-based access control (RBAC) system with hierarchical organization roles, resource-specific roles, and just-in-time (JIT) access management.

About This Lab Exercise

This application demonstrates a comprehensive authorization system implemented as part of Laboratory Exercise 3. The system provides:

- secure user authentication with two-factor authentication (2FA) and email verification,
- role-based access control (RBAC) with hierarchical organization-level roles (ORG_ADMIN, ORG_USER, ORG_GUEST),
- resource-specific roles (USER_READER, USER_WRITER) with least privilege principle,
- just-in-time (JIT) access management with automatic expiration and revocation,
- permission-based access control for different sections of the application,
- and secure user management with CRUD operations protected by role permissions.

Your Account Information

Username: mpet

Email: mpet@test.com

Registered on: November 11, 2025 at 4:48 PM

[Request Reader Permission](#)

[Request Writer Permission](#)

Registered Users

ID	Username	Email	Registered
14	admin	admin@admin.com	Dec 14, 2025
9	vbla	vbla@test.com	Nov 11, 2025
7	ilat	ilat@test.com	Nov 11, 2025
6	mpet	mpet@test.com	Nov 11, 2025

Total users: 4

© 2025 User Authentication System - Laboratory Exercise 3

USER_WRITER

Auth System

Welcome, mpet! [Logout](#)

Role-Based Access Control and Authorization System

Lab Exercise 3, made by Martina Petkovska - 223313, that implements a complete role-based access control (RBAC) system with hierarchical organization roles, resource-specific roles, and just-in-time (JIT) access management.

About This Lab Exercise

This application demonstrates a comprehensive authorization system implemented as part of Laboratory Exercise 3. The system provides:

- secure user authentication with two-factor authentication (2FA) and email verification,
- role-based access control (RBAC) with hierarchical organization-level roles (ORG_ADMIN, ORG_USER, ORG_GUEST),
- resource-specific roles (USER_READER, USER_WRITER) with least privilege principle,
- just-in-time (JIT) access management with automatic expiration and revocation,
- permission-based access control for different sections of the application,
- and secure user management with CRUD operations protected by role permissions.

Your Account Information

Username: mpet

Email: mpet@test.com

Registered on: November 11, 2025 at 4:48 PM

[Request Reader Permission](#)

[Request Writer Permission](#)

Registered Users

[Create user](#)

ID	Username	Email	Registered	Actions
14	admin	admin@admin.com	Dec 14, 2025	Edit Delete
9	vbla	vbla@test.com	Nov 11, 2025	Edit Delete
7	ilat	ilat@test.com	Nov 11, 2025	Edit Delete
6	mpet	mpet@test.com	Nov 11, 2025	Edit Delete

Total users: 4

© 2025 User Authentication System - Laboratory Exercise 3

5. Тестирање на CRUD операции

- Дозволено само за админ и за корисник(ORG_USER) со USER_WRITER JIT привремена улога.

Edit User

Username

Email

Update Cancel

Create User

Username

Email

Password

Create Cancel

Your Account Information

localhost:8080 says
Are you sure you want to delete this user?

OK Cancel

Username: admin
Email: admin@admin.com
Registered on: December 14, 2025 at 3:31 PM
[Request Reader Permission](#) [Request Writer Permission](#)

Registered Users

ID	Username	Email	Registered	Actions
14	admin	admin@admin.com	Dec 14, 2025	Edit Delete
9	vbla	vbla@test.com	Nov 11, 2025	Edit Delete
7	ilat	ilat@test.com	Nov 11, 2025	Edit Delete
6	mpet	mpet@test.com	Nov 11, 2025	Edit Delete

Total users: 4 [Create User](#)