

# Задача форма за регистрация

Технологии, използвани за създаването на уеб-приложение тип клиент-сървър:

- за back-end: Java 20 Gradle
  - библиотека Jakarta Servlet, с която мога да съзdam WebServlet, което позволява комуникация между браузера и сървъра
  - библиотека java.sql, която ми позволява да свържа приложението си с база данни и да извършвам заявки в нея
  - библиотека java.io.IOException, с която обработвам грешки
  - org.mindrot.jbcrypt.BCrypt се използва за криптиране на потребителските пароли, за да не се пазят в базата като чист текст - подобрява сигурността на приложението ми
- за front-end:
  - HTML и CSS за създаване на страницата, която се визуализира в браузера
  - JavaScript за Captcha и за проверка на статус от URL адреса
- за база данни: MySQL

Функционалности:

- валидация на данните: в клас Register извършвам проверка дали всички полета са запълнени, ако не са - връща грешка, а във front-end частта използвам формулляр, който проверява дали имейла е в правилния формат, както и скрива паролата
- записване на данните в релационна база данни:
  - имам клас MySQLDataSource, който указва на програмата ми къде да достъпи базата данни
  - клас DataBase, в който съм написал SQL заявка, създаваща таблица "Users" ако такава не съществува
  - в класа UserDaoImpl се осъществява връзка с базата данни и чрез SQL заявки се записват новите потребители или се променят атрибутите им
- За логин имам клас, който проверява дали този акаунт съществува и ако да - пренасочва към профил страницата, ако не - дава грешка за невалидни данни
- Имам страница наречена Profile, където потребителят може да промени избрани атрибути на профила си, като отново се минава проверка и при грешка се връщат errors; в тази страница се намира логаут бутоонът, който връща към страницата за логин
- Създад съм Unit тестове за всички функционалности на класа UserService, които са и всички функционалности на програмата ми
- CAPTCHA - тя се намира във front-end частта в register.html и представлява кратък JS скрипт, който създава две случаини числа и кара потребителя да ги събере и проверява дали отговора на потребителя съвпада с правилния отговор

Файлове и кой за какво се отнасят:

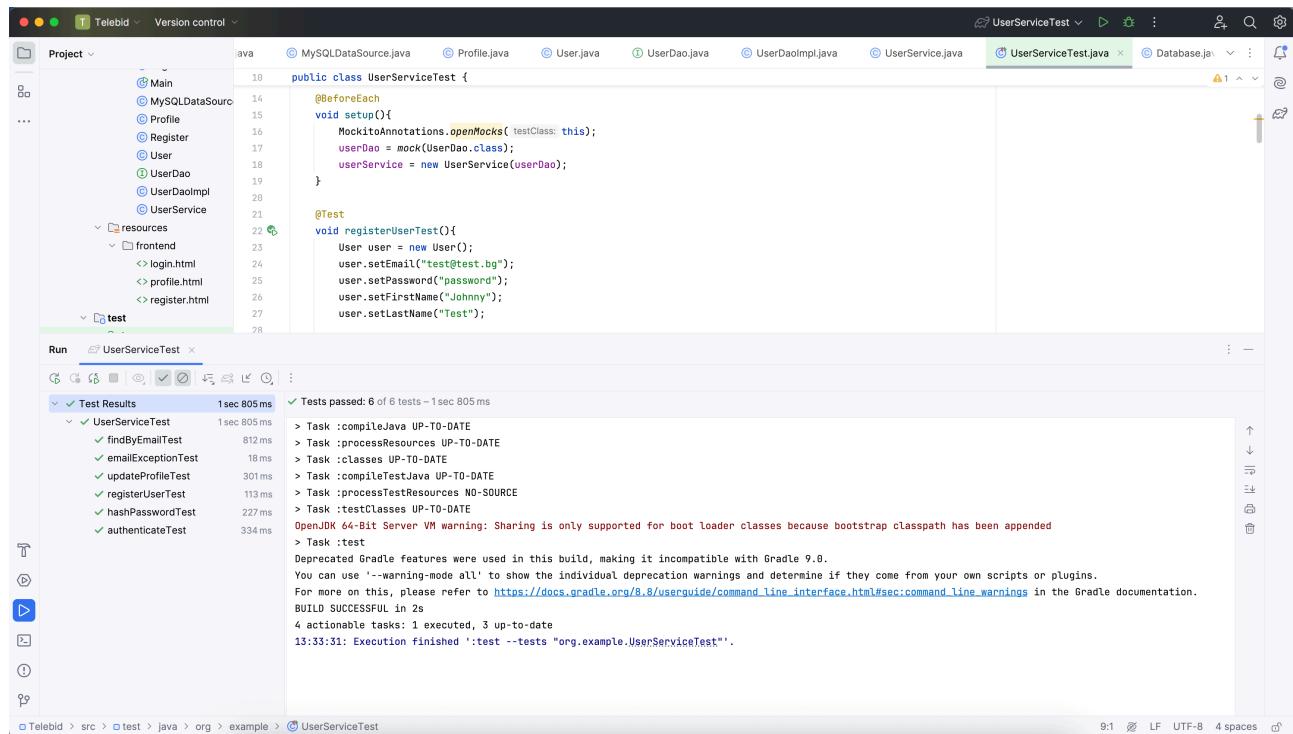
main/java/org.example:

- Database - създаване на таблица за потребители в базата данни
- Login - логин
- Main - от тук пускаме програмата
- MySQLDataSource - връзка с базата данни
- Profile - промяна на потребителските данни
- Register - регистрация на нов потребител
- User - тук дефинираме потребителя и неговите атрибути
- UserDao - интерфейс, който указва какви функции ще има User
- UserDaoImpl - имплементация на гореспоменатите функции
- UserService - функциите, които програмата изпълнява

main/java/resources/frontend - Тук са файловете от front-end частта на приложението:

- login.html
- profile.html
- register.html

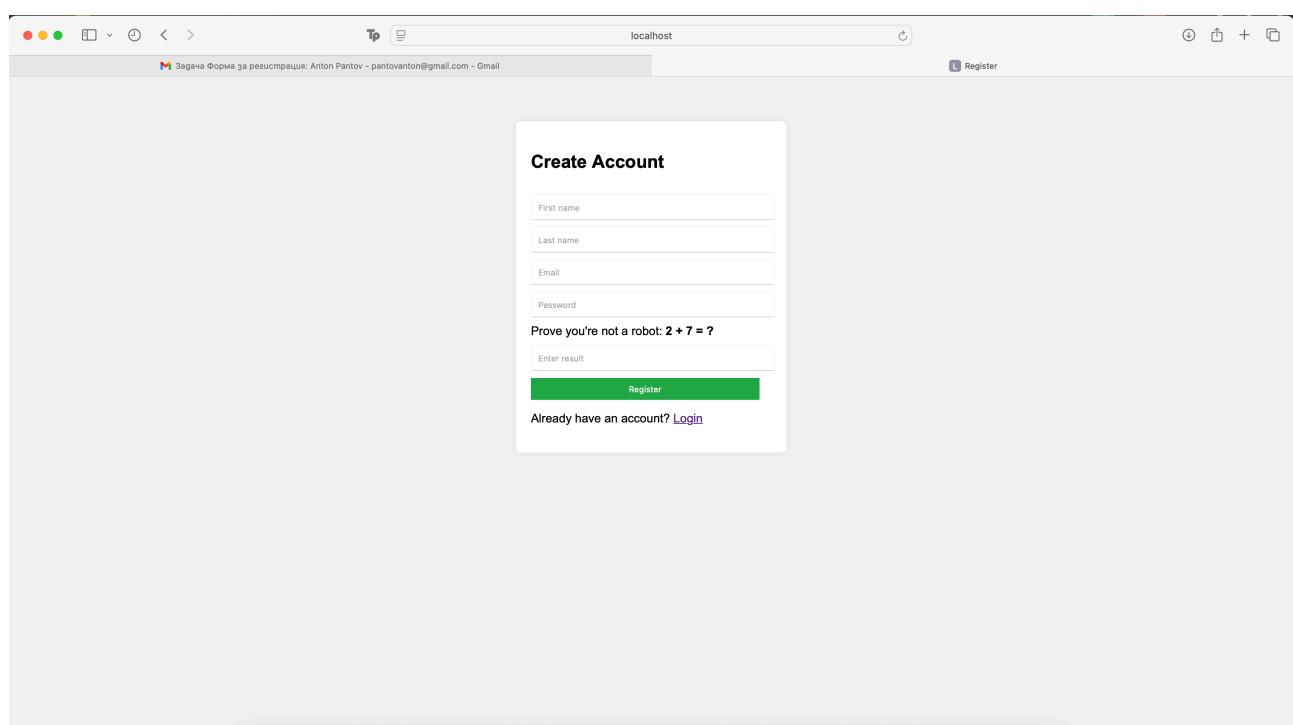
test/java/org.example/UserServiceTest - Тук се намират Unit тестовете, които ще прикача като скрийншот:



The screenshot shows an IDE interface with the following details:

- Project View:** Shows the project structure with packages like Main, MySQLDataSource, Profile, Register, User, UserDao, UserDaolmpl, and UserService. Under resources/frontend, there are files login.html, profile.html, and register.html.
- Code Editor:** The UserServiceTest.java file is open, showing Java code for testing the UserService. It includes a setup method using MockitoAnnotations and a registerUserTest method.
- Run View:** The Test Results window shows 6 tests passed in 1 second and 805 ms. The tests include findByEmailTest, emailExceptionTest, updateProfileTest, registerUserTest, hashPasswordTest, and authenticateTest.
- Bottom Status:** Shows the current file path as Telebid > src > test > java > org > example > UserServiceTest, and the terminal output indicating a successful build.

Прилагам и скрийншотове на User Interface-а:



Задача Форма за регистрация: Anton Pantov - pantovanton@gmail.com - Gmail

localhost

Login

Don't have an account? [Register](#)

Задача Форма за регистрация: Anton Pantov - pantovanton@gmail.com - Gmail

localhost

Profile

Update Profile

New first name (optional)

New last name (optional)

New email (optional)

New password (optional)

Profile updated successfully