**ТЕМА:**

**„Борд със задачи, които трябва да се изпълнят**

**по време на курс по WEB.“**

**Изготвил:**

Антония Тодорова

a.t.todorova97@gmail.com

**СЪДЪРЖАНИЕ**

TEMA - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - ( 3 )

УСЛОВИЕ - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - ( 3 )

ВЪВЕДЕНИЕ - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - ( 3 )

ТЕОРИЯ - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - ( 3 )

ИЗПОЛЗВНИ ТЕХНОЛОГИИ - - - - - - - - - - - - - - - - - - - - - - - - ( 3 )

ИНСТАЛАЦИЯ И ТАСТРОЙКИ - - - - - - - - - - - - - - - - - - - - - - - ( 4 )

КРАТКО РЪКОВОДСТВО НА ПОТРЕБИТЕЛЯ - - - - - - - - - - - - ( 5 )

ПРИМЕРНИ ДАННИ - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - ( 10 )

ОПИСАНИЕ НА ПРОГРАМНИЯ КОД

Php - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - ( 11 )

Html - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - ( 12 )

Css - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - ( 13 )

ПРИНОСИ НА СТУДЕНТА, ОГРАНИЧЕНИЯ И ВЪЗМОЖНОСТИ ЗА БЪДЕЩО РАЗВИРИЯ - - - - - - - - - - - - - - - - - - - - - - - - - - - ( 15 )

КАКВО НАУЧИХ - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - ( 15 )

ИЗПОЗЛВАНИ ИЗТОЧНИЦИ - - - - - - - - - - - - - - - - - - - - - - - ( 15 )

**Тема:**

Борд със задачи, които трябва да се изпълнят по време на курса по WEB.

**Условие:**

Да се направи проект, който дава възможност на преподавателите да възлагат задачи, вписвайки условията и крайния срок, а на студентите да изпращат решенията им и да ги маркират като завършени.

**Въведение:**

Целта на настоящия проект е да улесни както преподавателите, така и студентите в университета. Възможно е да се внедри във всеки един предмет. Предоставя функционалност за лесно въвеждане на задачи от страна на преподавателя. Той може да избере както студента, на когото да възложи дадена задача (може да избере всички студенти), така и дата, която определя крайния срок за предаване на заданието. От друга страна студента лесно може да проследява статуса на всички задания, които са му възложени, да следи крайните срокове и да предаде решение, когато е завършил работата си.

**Теория:**

Цялата нужна информация за проекта може да се намери в този файл.

**Използвани технологии:**

За да бъде създадено приложението са използвани следните технологии:

* visual studio code – средата, на която е написан кода
* xAmpp – приложение, позволяващо свързване с базата
* phpMyAdmin – интерфейс, предоставящ възможност за лесно и безпроблемно боравене с базата

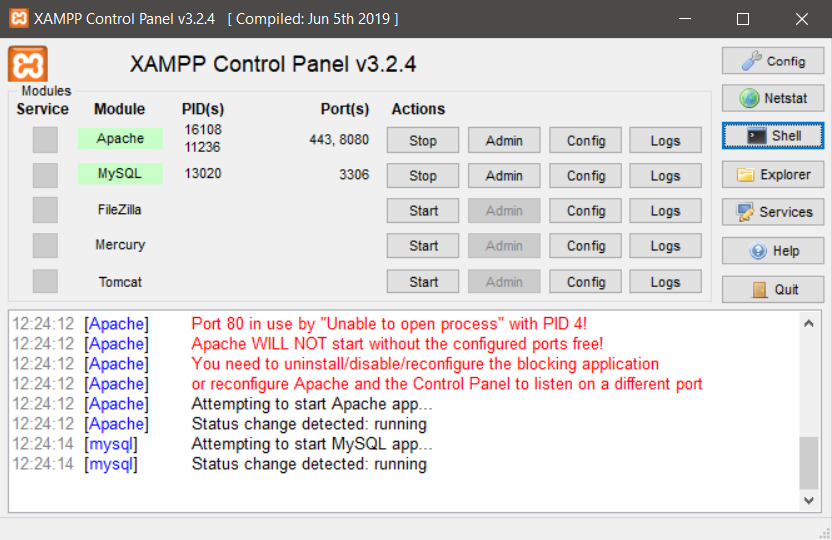
Използваните езици в проекта са:

* php – за създаването на бизнес логиката и комуникация с базата, предоставяйки основната функционалност на приложението
* html – за създаване на визуализацията и оформяне на основните функционалности, които предизвикват взаимодействие с базата
* css – за стилизация и оформяне на визуализацията на проекта, предавайки му по интуитивен и красив интерфейс

**Инсталация и настройки:**

За да се стартира приложението не се изискват много специални настройки. Най-важното е на компютъра да има инсталиран xAmpp, който работи със стандартните настройки.

Трябва да се стартират Apache и MySQL сървисите от xAmpp-а. Когато това стане, съответните модули се оцветяват в зелено, което е индикация че са успешно стартирани и до тях се показват портовете, на които работят. Там се намира съответния порт, на който ще се викне началната страница на приложението.



При първоначално стартиране на сървисите, трябва да се import-не базата в MySQL сървиса. За целта се влиза в Shell-а и се пише командата „mysql –u root –p project < dbproject.sql“, което import-ва цялата база и таблици, чиито dump файл с името dbproject.sql се намира в DatabaseDump папката на проекта. Веднъж направена тази стъпка, остава само да се провери дали базата се е import-нала правилно, чрез влизане в phpMyAdmin интерфейса, за достъп и лесно боравене с базата.

Остава единствено да се проверят настройките на самия проект дали съвпадат с настройките на xAmpp-a. Ако не съвпадат, то във файла с име connect.php трябва да се променят следните неща:

* $host = “localhost” – показва на кой хост ще работи приложението
* $db\_user = “root” – показва името, което предоставя достъп до базата
* $db\_password = “” – показва паролата, за достъп до базата
* $db\_name = “project” – показва името на базата с таблиците, които използва проекта (така е кръстена базата в dump файла)

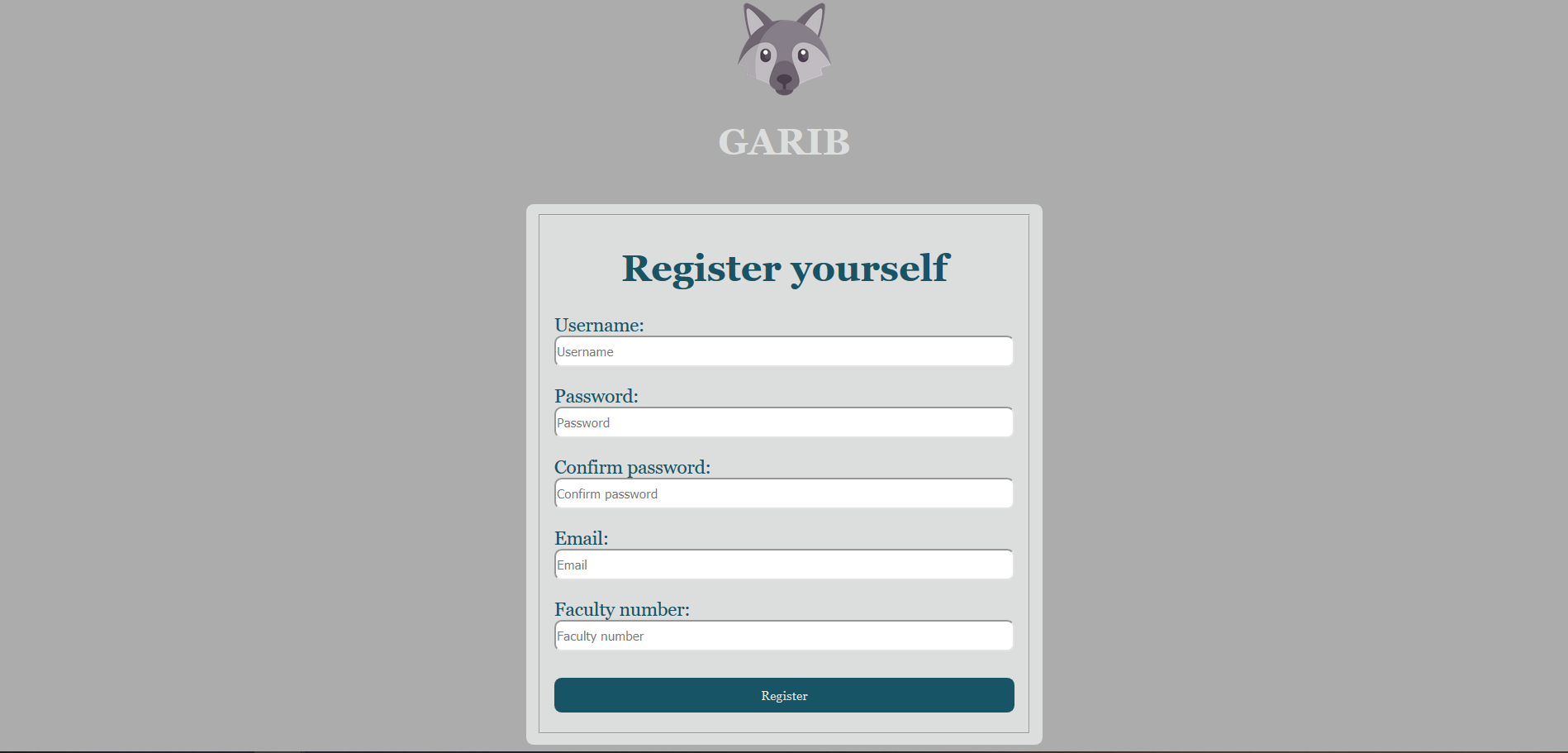
След изпълнението на всички тези стъпки приложението е готово за използване. Очаква се всичко да работи безпроблемно.

**Кратко ръководство на потребителя:**

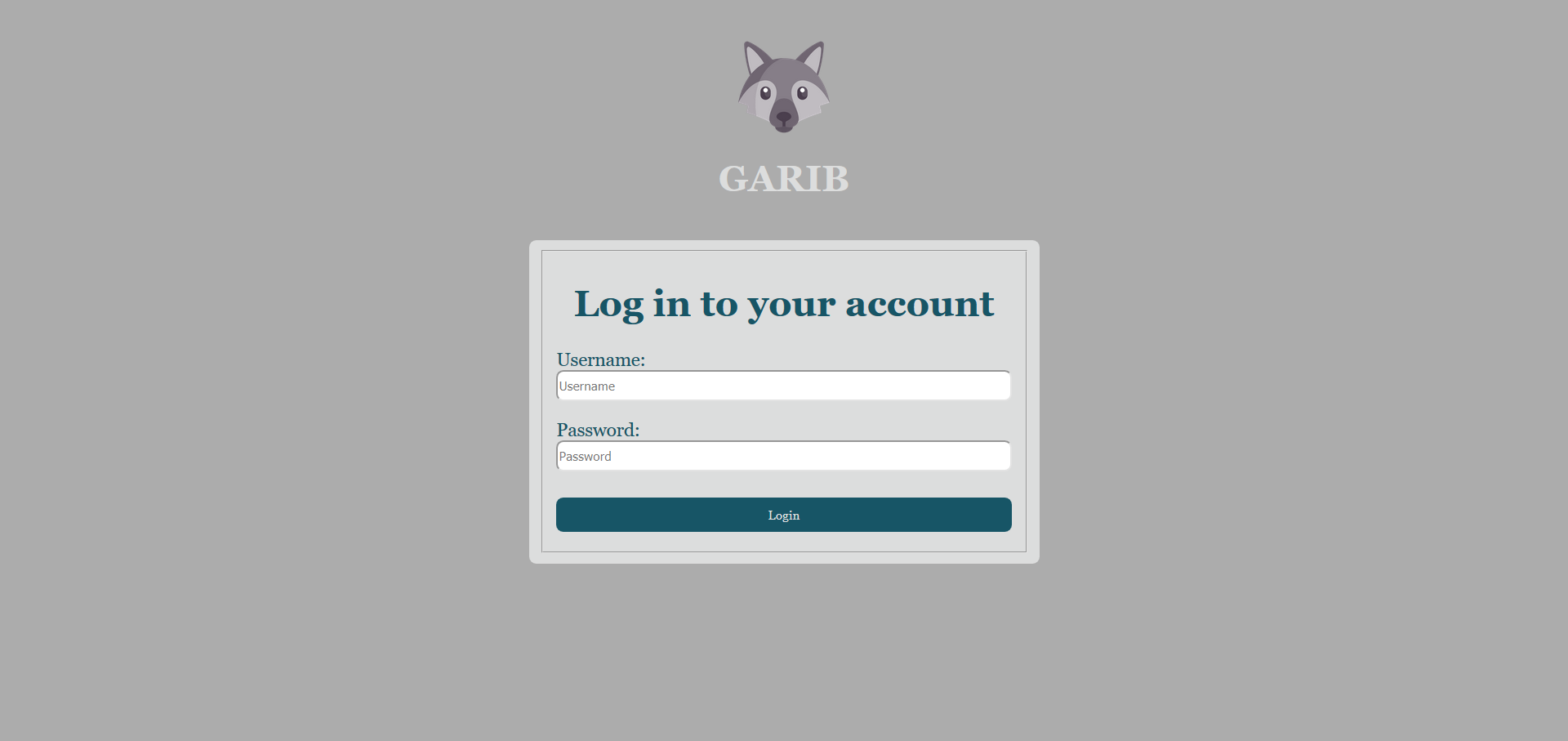
Приложението е доста интуитивно и всеки може да се научи да го използва за изключително кратко време.

Има две главни роли, които са преподавател и студент. Те имат малки разлики във функционалностите, предоставени им от приложението.

При първоначално влизане в приложението, всеки студент трябва да се регистрира. Относно преподавателите, всички са регистрирани автоматично.



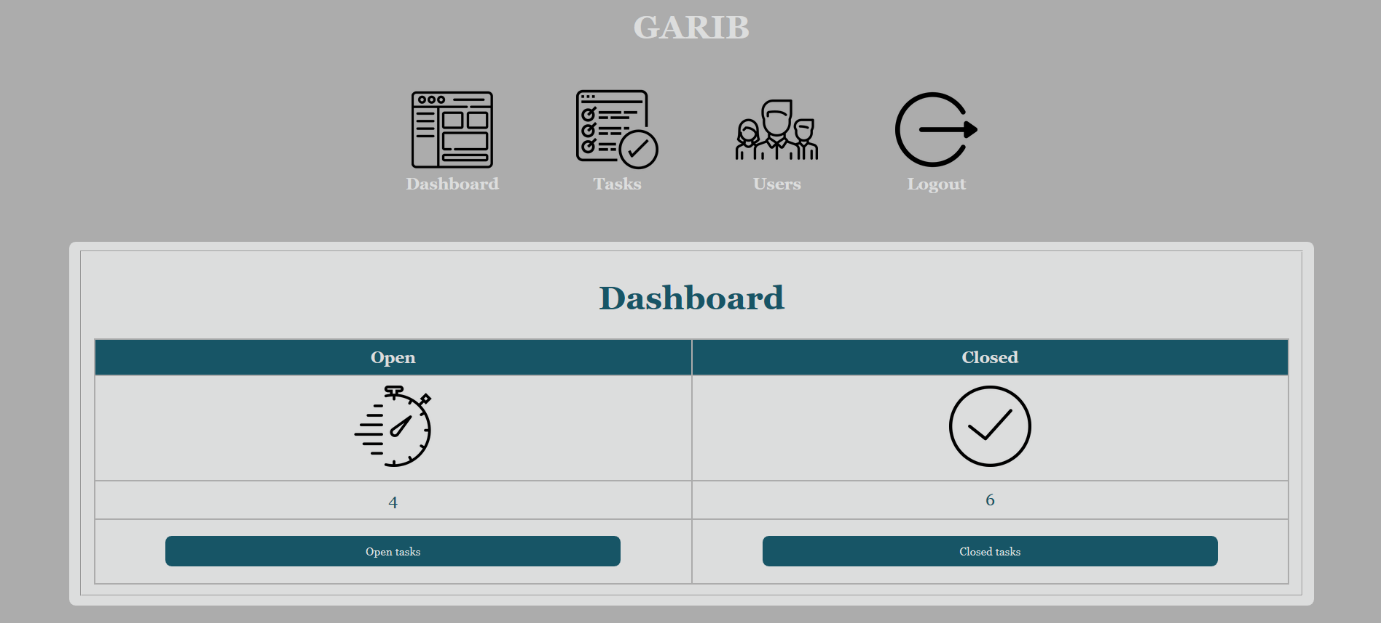
След успешна регистрация се дава възможност за влизане в приложението.



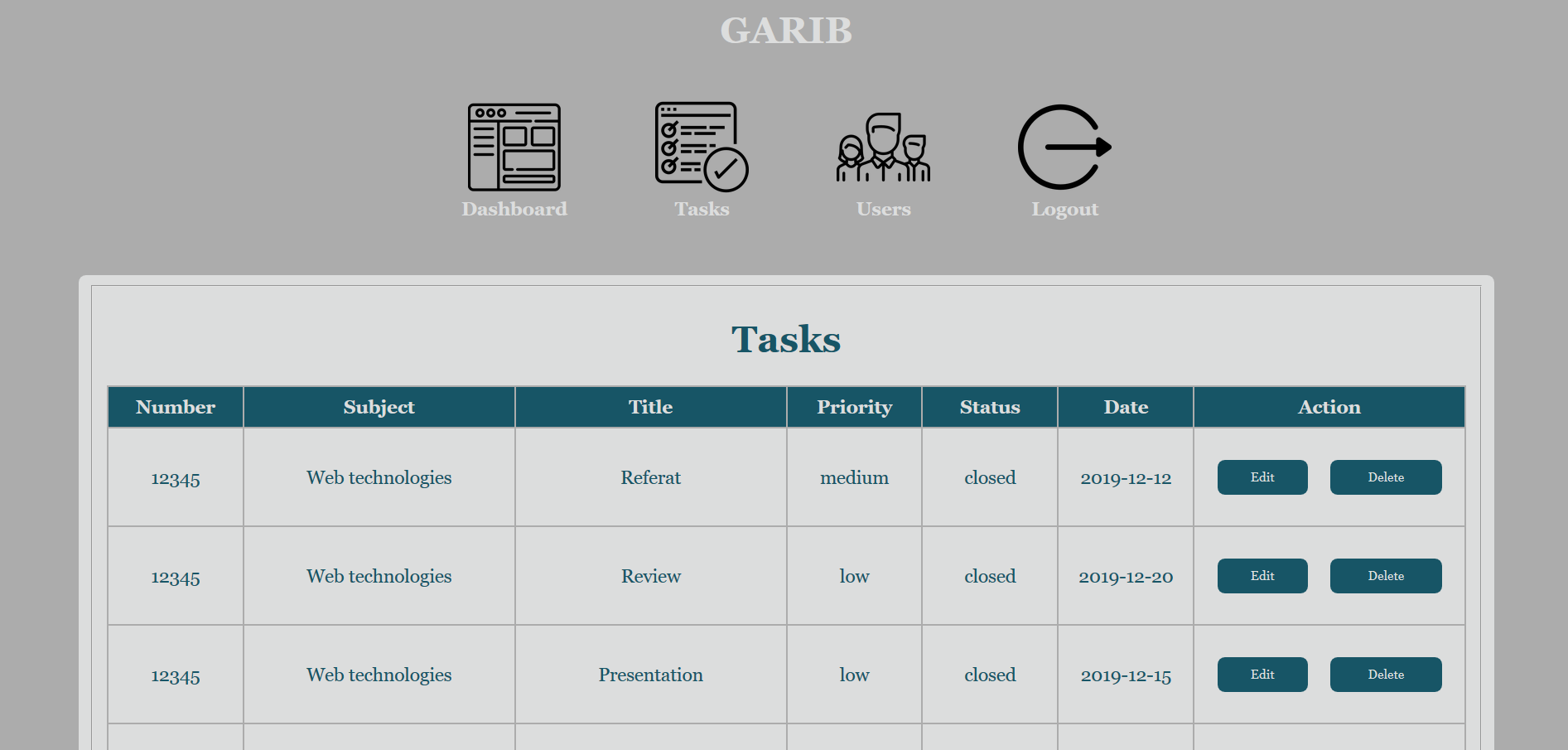
Зависимост от това какви права има съответния потребител се отваря различна начална страница. Преподавателя има всички функционалности, показани по-долу. Студента вижда Dashboard, Tasks и Logout бутоните, но няма видимост до Users, тъй като този бутон дава достъп до потребителите в системата, които могат да се изтриват единствено от администраторите. Всяка една от функционалностите ще бъде разгледана поотделно.



При натискане на Dashboard, се отваря табло показващо колко на брой отворени и затворени задачи има. Преподавателите виждат броя на всички задачи, докато студентите виждат броя само на собствените си такива. Когато се натиснат бутоните Open tasks или Closed tasks се достига съответно до отворените или затворените задачи.

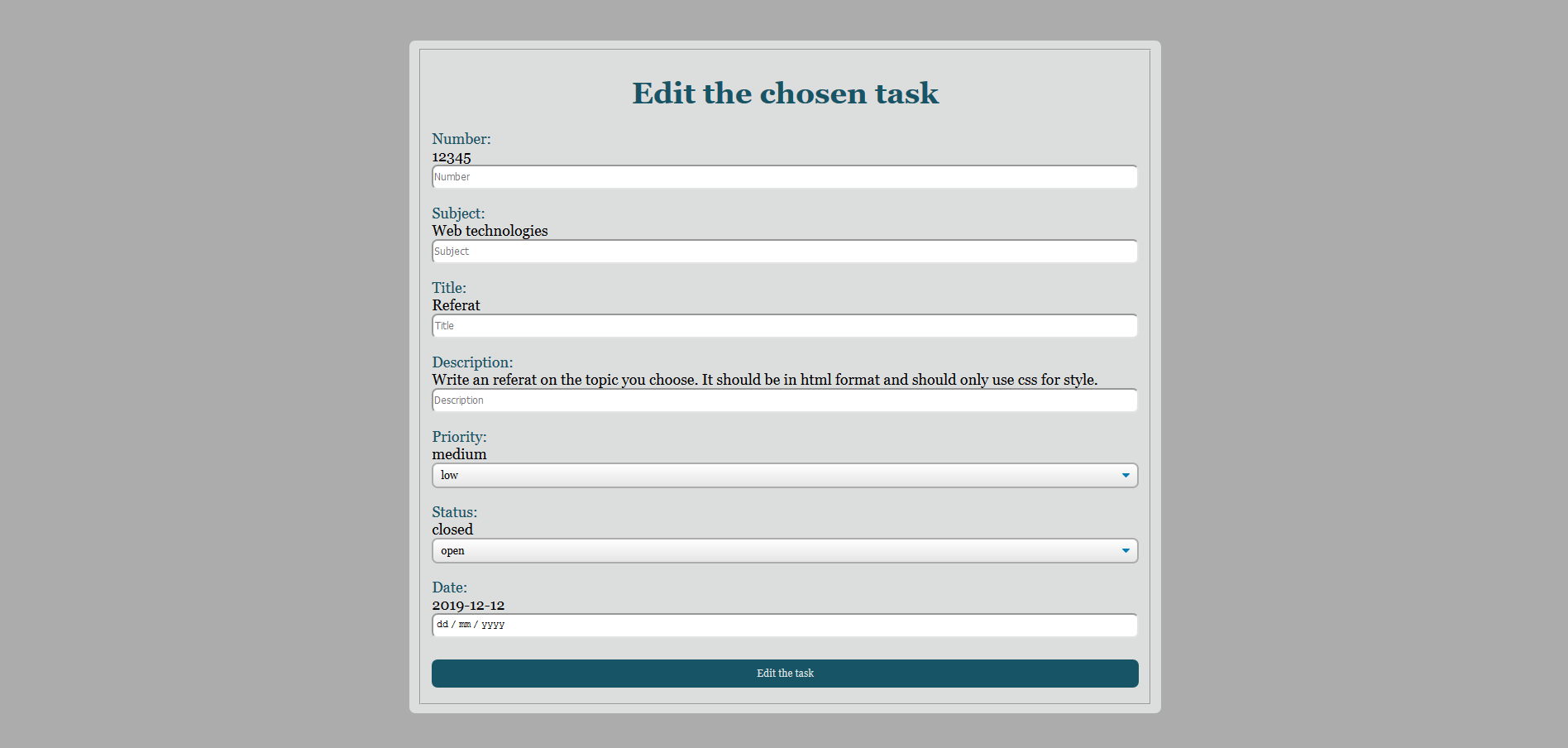


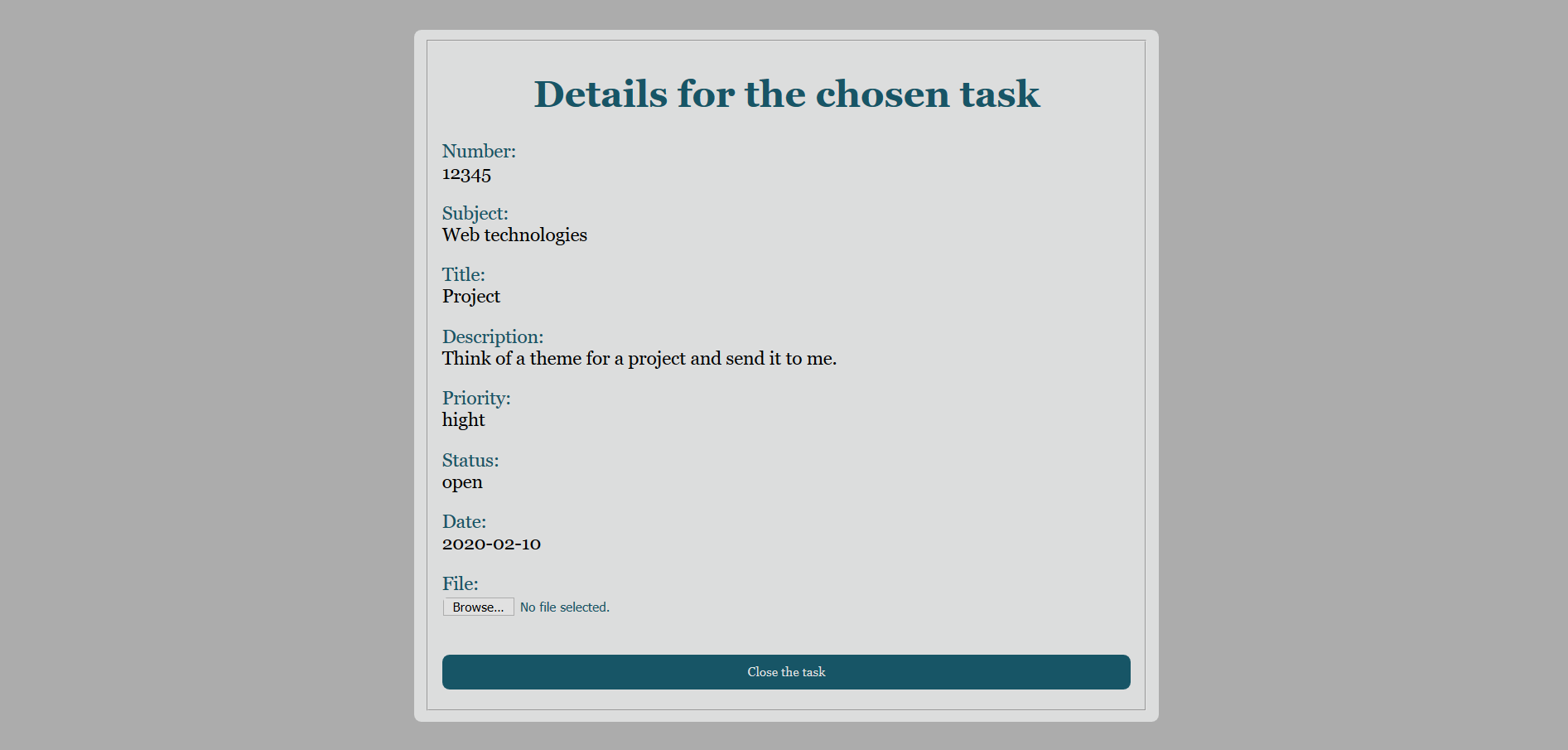
Когато се натисне бутона Tasks се отваря таблица, показваща подробна информация за всяка една задача. Ако потребителя има преподавателски права вижда задачите за всички студенти, а ако потребителя има студентски права вижда само своите задачи. Предоставена е информация за факултетния номер да студента, предмета и заглавието на задачата, с какъв приоритет е, в какъв статус се намира и коя е дата за крайния срок на предаване. Съответно преподавателите могат да редактират всяка една задача, а студентите да я преглеждат и предават решение.





Когато се натиснат Edit или View бутоните се отваря съответния модел за избраната задача. При Edit се показва информация за досегашните данни, като във всяко поле може да се въведе нова информация. Това позволява на преподавателя да промени информацията за всяка задача която желае. При View се показва информация за самата задача. Там има и Description поле, което задава изискванията. Най-отдолу има поле за качване на решение и съответно при натискане на Close task бутона, задачата минава в статус closed. Веднъж затворена задача, тя никога не може да бъде отворена отново за предаване на друго решение. Могат единствено да се виждат данните за нея.

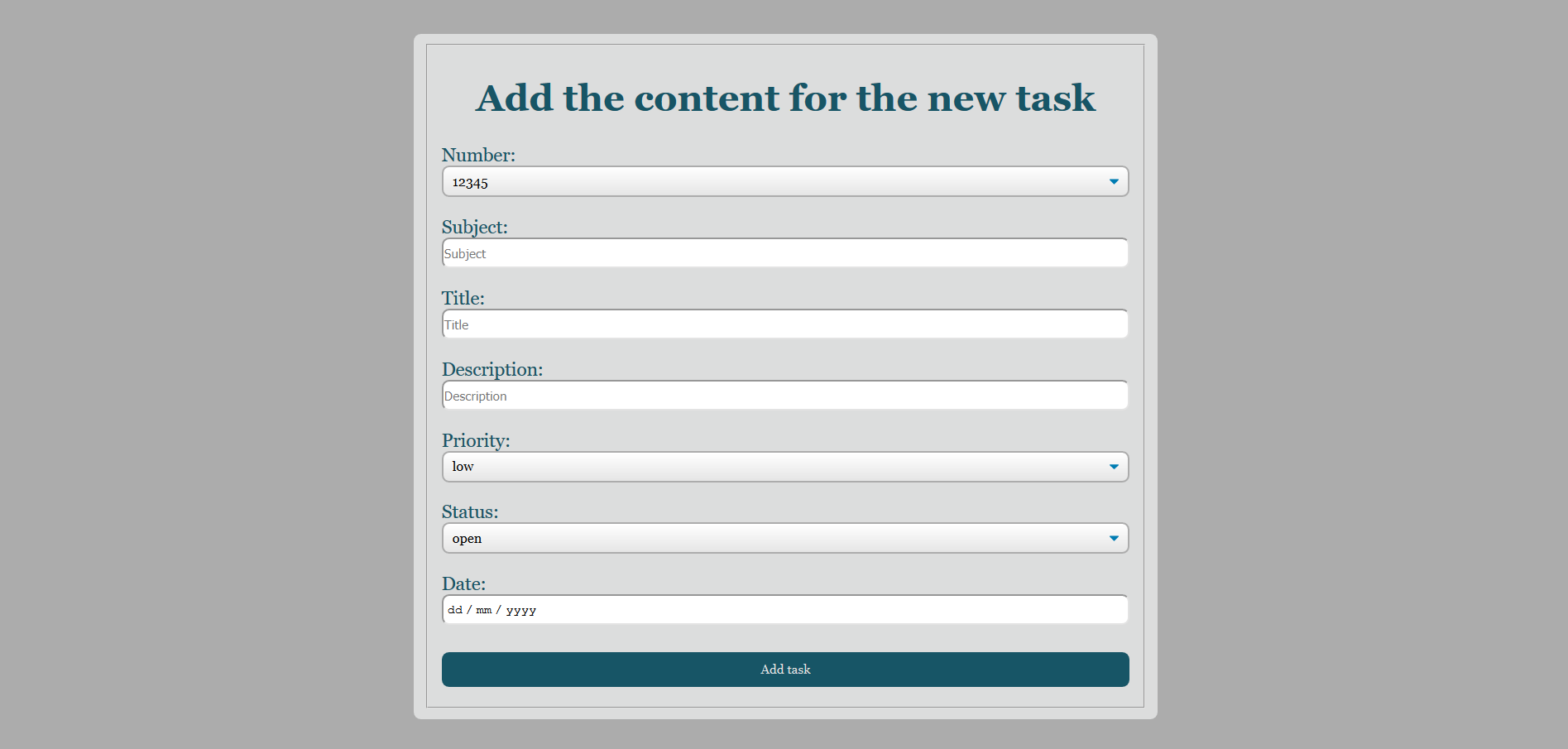




Всеки преподавател има възможност за въвеждане на нова задача. Това става чрез натискане на Add task бутона, намиращ се най-отдолу в таблицата на Tasks.



Показва се форма за въвеждане на нова задача. Предоставя се възможност да се избере конкретен студент, на когото да се възложи задачата или да се изберат всички студенти, които ще получат еднакви условия на задачата.

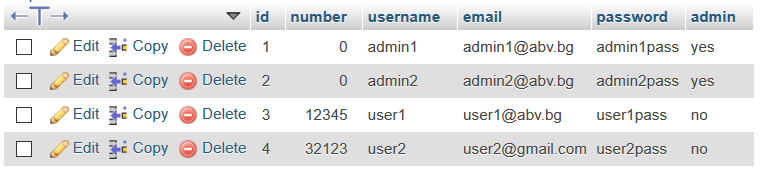


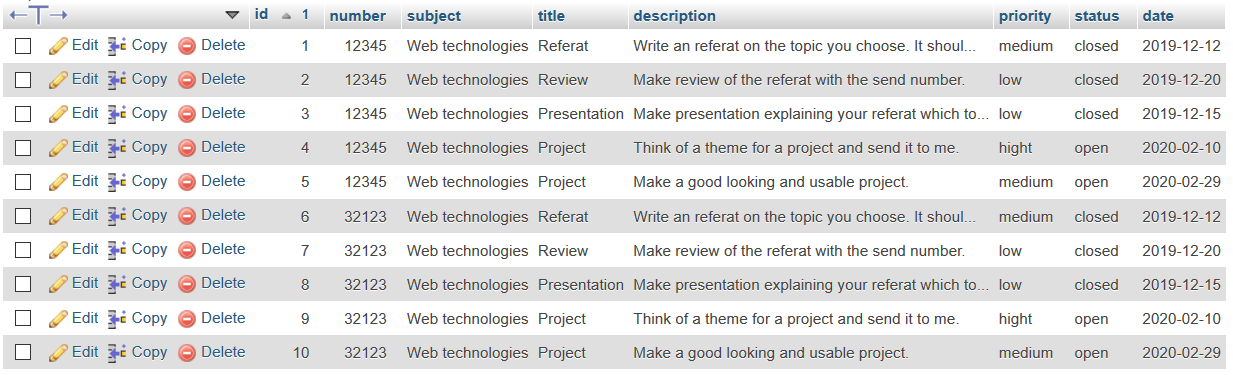
Гореспоменатите неща са основните функционалности на проекта. Веднъж изпробвани, се придобива усет към лесния и интуитивен интерфейс.

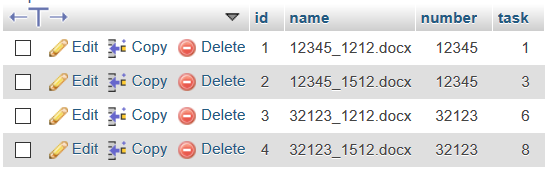
**Примерни данни:**

Данните, които се използват от проекта се записват в три различни таблици. Те са следните:

* users – притежава данни за потребителите, които използват приложението, като факултетен номер, имейл, потребителско име, парола
* tasks – притежава данни за задачите, възложени на съответните студенти, като факултетен номер, предмет, заглавие, описание, приоритет, статус и дата
* files – притежава данни за качените решения от студентите, като факултетен номер, име на файла и номер на задачата, на която съответства решението







**Описание на програмния код:**

Приложението използва .php, .html и .css файлове. Те се намират в главната папка на проекта.

* php – задава самата функционалност, като си комуникира с базата. Прави се сесия, която показва дали потребителя е влязъл в системата, осъществява се свързване с базата, изпращат се заявки и се получава отговор с поисканите данни, правят се проверки за данните и на база техните отговори се препраща към други файлове от приложението.

Пример за такъв файл е следния (login.php - изпълнява login-ването в системата):

<?php

    session\_start();

    require\_once "connect.php";

    $connection = new mysqli($host, $db\_user, $db\_password, $db\_name);

    if($connection->connect\_errno!=0){

        echo "Error: ".$connection->connect\_errno . "<br>";

        echo "Description: " . $connection->connect\_error;

        exit();

    }

    else{

        $username = $\_POST['username'];

        $password = $\_POST['password'];

        $sql = "SELECT \* FROM users

WHERE username='$username'

AND password='$password'";

        if($result = $connection->query($sql)){

            $usersCount = $result->num\_rows;

            if($usersCount>0){

                $\_SESSION['logged-in'] = true;

                $row = $result->fetch\_assoc();

                $user = $row['username'];

                $result->free\_result();

                $\_SESSION['user'] = $user;

                unset($\_SESSION['loginError']);

                $roleSQL = "SELECT \* FROM users

WHERE username='$username'";

                $resultRole = $connection->query($roleSQL);

                $rowRole = $resultRole->fetch\_assoc();

                if($rowRole['admin'] === 'yes'){

                    header('Location: homeAdmin.html');

                }

                else {

                    header('Location: homeUser.html');

                }

            }

            else{

                $\_SESSION['loginError'] =

'<span class="error-msg">Invalid inputs.</span>';

                if(isset($\_SESSION['loginError'])){

                    echo $\_SESSION['loginError'];

                }

                header('Location: login.html');

            }

        }

        $connection->close();

    }

    if(isset($\_SESSION['loginError'])){

        echo $\_SESSION['loginError'];

        exit();

    }

?>

* html – създава визуализацията и оформя целия интерфейс на приложението. Чрез използване на тагове, които предоставя този език се структурира целия дизайн, върху който ще се направи функционалността.

Пример за такъв файл е следния (registration.html – визуализира формата за регистрация на нов потребител в системата):

<!doctype html>

<html>

    <head>

        <meta charset = "UTF-8">

        <title> Registration </title>

        <link rel = "stylesheet" href = "login.css">

    </head>

    <body>

        <header id = "header">

            <div>

                <img class = "icon" id = "wolf"

src = "Pictures/wolf.png">

                <h1 id = "garib"> GARIB </h1>

            </div>

        </header>

        <form method = "POST",  action = "registration.php">

            <fieldset>

                <h1> Register yourself </h1>

                <p>

                    <label for = "username"> Username: </label>

                    <input type = "text" name = "username"

placeholder = "Username">

                </p>

                <p>

                    <label for = "password"> Password: </label>

                    <input type = "password" name = "password"

placeholder = "Password">

                </p>

                <p>

                    <label for = "confirmPassword">

Confirm password:

</label>

                    <input type = "password" name = "confirmPassword"

placeholder = "Confirm password">

                </p>

                <p>

                    <label for = "email"> Email: </label>

                    <input type = "email" name = "email"

placeholder = "Email">

                </p>

                <p>

                    <label for = "facultyNumber">

Faculty number:

</label>

                    <input type = "text" name = "facultyNumber"

placeholder = "Faculty number">

                </p>

                <button class = "Button" type = "submit"

value = "Register">

Register

</button>

            </fieldset>

        </form>

    </body>

</html>

* css – стилизира и оформя визуализацията на проекта, предавайки му по интуитивен и красив интерфейс. Той задава различни стойности на всеки един елемент, като оказва размера, цвета, позицията и още много други неща.

Пример за такъв файл е следния (login.css – осъществява стилизацията на формата за логване на потребител в системата):

body {

    background-color: rgb(172, 172, 172);

    font-family: Georgia, 'Times New Roman', Times, serif;

    margin: 20px 40px 20px 40px;

    font-size: 18px;

    color: #175566;

}

h1 {

    text-align: center;

    color: #175566;

}

#garib {

    color: rgb(220, 221, 221);

}

form {

    max-width: 500px;

    margin: 40px auto;

    padding: 10px 10px;

    background: rgb(220, 221, 221);

    border-radius: 7px;

}

.formButton{

    max-width: 500px;

    margin: 0px auto;

    padding: 0px 0px 0px 0px;

    background: rgb(220, 221, 221);

    border-radius: 7px;

}

input {

    width: 100%;

    height: 30px;

    border-radius: 7px;

}

textarea {

    width: 100%;

    height: 80px;

}

.Button {

    background-color: #175566;

    color: #E1E2E2;

    border-radius: 7px;

    padding: 10px 10px;

    margin: 8px 0;

    border: none;

    cursor: pointer;

    width: 100%;

    font-family: Georgia, 'Times New Roman', Times, serif;

}

Всички тези файлове си взаимодействат помежду си, оформяйки цялостното приложение. Това става когато файловете се извикват помежду си, в зависимост каква функционалност им е нужна. За да изглежда добре и да функционира правилно проекта, трябва да се постигне хармония между всички файлове.

При първоначално стартиране на приложението се извивка файла start.html, който в последствие ще си взаимодейства с всички останали функционалности, превръщайки приложението във верига от файлове.

**Приноси на студента, ограничения и възможности за бъдещо развитие:**

Проектът е направен от един студент, чиито данни са посочени в заглавната страница да документацията. Тъй като няма предишен опит в web програмирането, всички функционалности са изпълнени с базови знания и умения, придобити от курса в университета. Относно бъдещото развития – винаги има на къде повече човек да се усъвършенства и постоянно трябва да е готов и да има желание да обогати знанията си.

**Какво научих:**

Придобитите знания са свързани с изграждането на web приложение, усвояване на езиците php, html и css, боравене с база данни и използване на приложениe като xAmpp, което дава бърз и лесен достъп до базата данни с phpMyAdmin.

**Използвани източници:**

Всички използвани източници са предоставени от курса, като презентации и примерни файлове с решения от лекции и упражнения.