



ТЕХНИЧЕСКИ УНИВЕРСИТЕТ -
СОФИЯ ФИЛИАЛ ПЛОВДИВ

КУРСОВА РАБОТА по ПИ

НА

Веселин Руменов Станчев

Фак. № 614872

Специалност: Компютърни системи и технологии
Образователно-квалификационна степен: Магистър

Изготвил:
Веселин Руменов Станчев

Пловдив 2022

Приложение GNUThink

Моделът на проектиране на курсовата работа е open-source. Курсовата работа представлява уеб приложение, в което са реализирани знанията, получени по предмета Програмиране за Интернет. Те обхващат:

- HTML;
- CSS;
- JavaScript;
- Back-end server language - PHP;
- RegEx.

Темата е приложение подобно на системата за управление на учебния процес с отворен код Moodle.

Възможностите на приложението включват:

- логин и регистрация
- разделяне на достъпа на студент и преподавател
- добавяне на урок от страна на преподавател
- разглеждане на уроци от страна на студент
- използвана SQLITE база данни

Използваните програмни езици са PHP, HTML, CSS, JS.

За Front-End частта са използвани HTML,CSS,JS.

HTML е основният маркиращ език за изграждане на структурата на уеб страниците от които е изградено приложението.

В настоящото приложение чрез HTML са изградени потребителските форми за логин, регистрация, добавяне на уроци.

Форма за логин

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Login</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="stylesheet" type="text/css" href="styles.css" />
  </head>
  <body>
    <h1>Вход</h1>
    <form method="post" action="logincheck.php">
      <input type="text" name="uname1" placeholder="Enter
username:"></br>
      <input type="password" name="upass" placeholder="Enter
password:"></br>
      <input type="radio" name="status" id="T" value="Teacher">
        <label for="T">Teacher</label></br>
      <input type="radio" name="status" id="S" value="Student">
        <label for="S">Student</label></br>
      <input type="submit" value="Login" name="submit">
    </form>
  </body>
</html>
```

Форма за добавяне на урок

```
<h1>Добави нов урок</h1>
  <form method="post" action="cr_article.php" enctype="multipart/form-data">
    <input type="text" name="title"></br>
    <textarea name="content">
  </textarea></br>
    <input type="file" name="resources" id="resources"></br>
    <input type="submit" value="Upload file" name="submit">
  </form>
```

CSS е основният маркиращ език за изграждане на потребителския дизайн на уеб страниците от които е изградено приложението.

В настоящото приложение чрез CSS са изградени дизайна на страниците .

CSS Layout

```
* {  
  box-sizing: border-box;  
}  
  
body {  
  margin: 0;  
}  
  
/* Style the header */  
.header {  
  background-color: #f1f1f1;  
  padding: 20px;  
  text-align: center;  
}  
  
/* Style the top navigation bar */  
.topnav {  
  overflow: hidden;  
  background-color: #333;  
}
```

JS е интерпретируем език от високо ниво за изграждане на потребителско взаимодействие между уеб страниците и потребителя.

В настоящото приложение чрез JS е изградена валидацията на потребителските форми .

Валидация на потребителската форма за регистрация

```
<script>
    function validateform(){
        if(document.regform.uname.value==""){
            alert("Enter valid username");
            document.regform.uname.focus();
            return false;
        }
        if(document.regform.upass.value==""){
            alert("Enter valid password");
            document.regform.upass.focus();
            return false;
        }
    }
</script>
```

За база данни е избран SQLITE, заради това че е съвместим със различни езици за програмиране като C/C++, Python, PHP. Освен това отпада необходимостта от използване на допълнителен СУБД софтуер, като например PHPMYADMIN. Цялата база данни се съхранява в sqlite файл. Всички заявки към базата данни са параметъризирани с цел превенция от тип атака sql injection.

```
<?php
$stmt = $db -> prepare("INSERT INTO users (UNAME,UPASS,STATUS)
VALUES (:UNAME, :UPASS, :STATUS);");
?>
```

За работата на приложението са необходими таблиците USERS и LESSON, разгледани по-долу.

ТАБЛИЦА USERS

поле	тип	описание
ID	INTEGER	РКЕУ ИД НА ПОТРЕБИТЕЛ
UNAME	VARCHAR(20)	ПОТР ИМЕ
UPASS	VARCHAR(20)	ПАРОЛА
STATUS	VARCHAR(20)	ТИП ПОТРЕБИТЕЛ

Таблица USERS съхранява данните необходими за логин и разделяне на достъпа.

ТАБЛИЦА LESSONS

поле	тип	описание
ID	INTEGER	РКЕУ ИД НА ПОТРЕБИТЕЛ
UNAME	VARCHAR(20)	ПОТР ИМЕ
UPASS	VARCHAR(20)	ПАРОЛА
STATUS	VARCHAR(20)	ТИП ПОТРЕБИТЕЛ

PHP е скриптовия език който се използва за връзката между потребителските форми и избраната база данни. За всяка потребителска форма съществува обслужваща php страница.

Потребителските страници също са с разширение .php за да може да бъдат използвани php обслужващи фрагменти вътре в HTML.

PHP страница за регистрация

```
<?php
if($_POST){
    $uname1=$_POST['uname1'] ;
    $upass=trim($_POST['upass']) ;
    $encpass=md5($_POST['upass']);
    $status=$_POST['status'];
    try{

        include('conn.php');

        $stmt = $db -> prepare("INSERT INTO users
(UNAME,UPASS,STATUS) VALUES (:UNAME,:UPASS,:STATUS);");

        $stmt -> bindParam(':UNAME', $uname1, PDO::PARAM_STR);
        $stmt -> bindParam(':UPASS', $encpass, PDO::PARAM_STR);
        $stmt -> bindParam(':STATUS', $status, PDO::PARAM_STR);

        if( $stmt -> execute() ){
            header("Location:index.php");
        }

        $db = null;
    }
    catch (PDOException $e){
        echo $e->getMessage();
    }
}
?>
```

PHP страница за уроци

```
<?php
if($_POST){
    session_start();
    $title = $_POST['title'];
    $content = $_POST['content'];
    $resource=$_FILES['resources'];
    $file_name = $_FILES['resources']['name'];
    $file_size =$_FILES['resources']['size'];
    $file_tmp =$_FILES['resources']['tmp_name'];
    $author=$_SESSION['username'];

    try{
        $upld= move_uploaded_file($file_tmp,"resources/".$file_name);
        include('conn.php');

        /* Create a prepared statement */
        $stmt = $db -> prepare("INSERT INTO lesson (TITLE,CONTENT,RESOURCES,author)
VALUES (:TITLE, :CONTENT,:RESOURCES,:author);");

        /* bind params */
        $stmt -> bindParam(':TITLE', $title, PDO::PARAM_STR);
        $stmt -> bindParam(':CONTENT', $content, PDO::PARAM_STR);
        $stmt -> bindParam(':RESOURCES', $upld);
        $stmt -> bindParam(':author', $author);
        /* execute the query */
        if( $stmt -> execute() ){
            header("Location:inner.php");
        }

        /* close connection */
        $db = null;
    }
    catch (PDOException $e){
        echo $e->getMessage();
    }
}
?>
```


