

Dokumentacja programu



Projekt_Chmura

Secure File Cloud (SFC)

Legenda:

1.Nazwa Projektu

2.Wprowadzenie

3.Technologie

4.Struktura

1.Baza danych

2.Strona

3.CSS

4.Skrypty

5.Instalacja

6.Testy

7.Wymagania systemowe

8.Licencja

9.Autorzy

1.Nazwa Projektu

Projekt_Chmura

Secure File Cloud (SFC)

2.Wprowadzenie

Projekt_Chmura " Secure File Cloud (SFC)" jest systemem do przechowywania i zarządzania plikami online. Umożliwia użytkownikom logowanie, rejestrację, przesyłanie plików na serwer, ich pobieranie, usunięcie oraz udostępnianie innym użytkownikom. Dodatkowo oferuje funkcje zmiany języka interfejsu oraz motywu graficznego.

Projekt zakłada że następujące założenia zostaną spełnione:

- 1.logowanie/rejestracja (możliwość zarejestrowania się oraz logowania w celu dostania się do chmury)
- 2.odzyskiwanie hasła (możliwość zresetowania hasła w przypadku jego zapomnienia)
- 3.zapewnienie bezpieczeństwa przechowywanych danych (dane zaszyfrowane)
- 4.dodawanie,usuwanie,pobieranie plików (pliki można wrzucić, usunąć a także pobrać z chmury)
- 5.udostępnianie plików przez linka (dla każdego pliku można utworzyć link dzięki któremu każdy może pobrać plik)
- 6.tymczasowe blokowanie dostępu do konta przy paru błędnych próbach logowania (w przypadku nieudanych prób logowania konto zostaje zablokowane na 20 minut)
- 7.zarządzanie kontami userów + statystyki z poziomu konta admina (konto admina z którego będzie dostęp do zarządzania użytkownikami. Mowa tutaj o możliwości zwiększenia miejsca na dysku oraz o usunięciu użytkownika. Podgląd statystyk. Mowa tutaj o informacjach takich jak zajęte miejsce przez poszczególnych użytkowników, suma zajętego miejsca)
- 8.motyw jasny/ciemny (dwa motywy strony jasny i ciemny)
- 9.dwie wersje językowe (strona po polsku i ukraińsku)
- 10.wyszukiwarka plików (możliwość wyszukiwania konkretnych plików w chmurze)

3.Technologie

Apache 2.4.52

html/css/javascript

MySQL 8.0.36

PHP 8.1.2

PHPMailer 6.9.1

Python 3.12.3 (do testowania po stronie klienta)

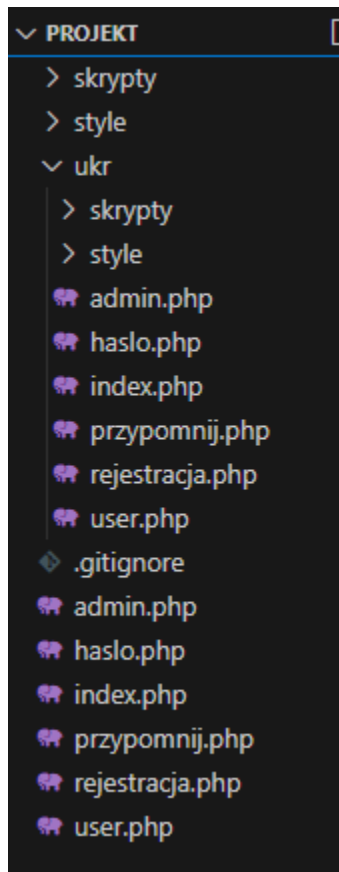
4.Struktura

Ułożenie katalogów:

Folder strony wygląda następująco:

W roocie znajdują się pliki z sekcji Strony(2) oraz foldery skrypty w których znajdują się pliki z sekcji Skrypty(4) i style w których znajdują się pliki z sekcji Style(3)

Ponadto w roocie znajduje się folder ukr w którym znajdują się wszystkie wcześniejsze pliki w tych samych miejscach ale przetłumaczone na język ukraiński



1.Baza danych

uzytkownicy: przechowuje dane użytkowników, w tym e-mail, hasło (zahashowane), typ użytkownika (admin/user), oraz dostępne miejsce na pliki.

```
CREATE TABLE IF NOT EXISTS `projekt_chmura`.`uzytkownicy` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `email` VARCHAR(64) NOT NULL,  
  `haslo` VARCHAR(128) NOT NULL,  
  `admin` BIT(1) NOT NULL DEFAULT 0,
```

```

`miejsce` INT(7) NOT NULL DEFAULT 50,
PRIMARY KEY (`id`),
UNIQUE INDEX `email_UNIQUE` (`email` ASC) )
ENGINE = InnoDB;

```

pliki: zawiera informacje o plikach przesłanych przez użytkowników, w tym nazwę pliku, jego zawartość, właściciela (powiązanie z użytkownikiem), oraz ewentualny kod do udostępnienia.

```

CREATE TABLE IF NOT EXISTS `projekt_chmura`.`pliki` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `nazwa` VARCHAR(100) NOT NULL,
  `plik` LONGBLOB NOT NULL,
  `kod` VARCHAR(20) NULL,
  `wlasiciel` INT NOT NULL,
  `bajty` INT(9) NOT NULL,
  PRIMARY KEY (`id`, `wlasiciel`),
  INDEX `fk_pliki_uzytkownicy_idx` (`wlasiciel` ASC) ,
  CONSTRAINT `fk_pliki_uzytkownicy`
    FOREIGN KEY (`wlasiciel`)
      REFERENCES `projekt_chmura`.`uzytkownicy` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

rejestracja: używana do przechowywania tymczasowych kodów rejestracyjnych i resetowania haseł.

```

CREATE TABLE IF NOT EXISTS `projekt_chmura`.`rejestracja` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `email` VARCHAR(64) NOT NULL,
  `kod` VARCHAR(20) NOT NULL,
  `czas` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

```

blokada: tabela do przechowywania adresów IP, które zostały zablokowane z powodu nieudanych prób logowania.

```

CREATE TABLE IF NOT EXISTS `projekt_chmura`.`blokada` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `ip` VARCHAR(15) NOT NULL,
  `czas` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

```

blokada2: tabela do przechowywania adresów IP, które zostały zablokowane z powodu za dużej liczby wysłanych wiadomości.

```

CREATE TABLE IF NOT EXISTS `projekt_chmura`.`blokada2` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `ip` VARCHAR(15) NOT NULL,
  `czas` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`))

```

```
ENGINE = InnoDB;
```

Eventy: zdarzenia odpowiedzialne za automatyzację usuwania rekordów z tabel blokujących dostęp po pewnym czasie.

Odblokowanie zablokowanego ip z tabeli blokada do logowania po upływie 20 minut:

```
--usuwanie zawartości z zablokowanymi ip
CREATE EVENT IF NOT EXISTS delete_expired_records_event
ON SCHEDULE EVERY 1 MINUTE -- Sprawdzanie co minutę, czy istnieją rekordy do usunięcia
DO
BEGIN
    DECLARE done BOOLEAN DEFAULT FALSE;
    DECLARE id_to_delete INT;
    DECLARE cur CURSOR FOR
        SELECT id
        FROM blokada
        WHERE TIMESTAMPDIFF(MINUTE, czas, NOW()) >= 20
        ORDER BY czas ASC;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

    OPEN cur;

    read_loop: LOOP
        FETCH cur INTO id_to_delete;
        IF done THEN
            LEAVE read_loop;
        END IF;

        DELETE FROM blokada WHERE id = id_to_delete;
    END LOOP;

    CLOSE cur;

END//
```

Usuwanie zawartości tabeli rejestracja po godzinie od pojawienia się rekordu:

```
--usuwanie zawartości z kodami do rejestracji i hasłem
CREATE EVENT IF NOT EXISTS delete_expired_records_event2
ON SCHEDULE EVERY 1 MINUTE -- Sprawdzanie co minutę, czy istnieją rekordy do usunięcia
DO
BEGIN
    DECLARE done BOOLEAN DEFAULT FALSE;
    DECLARE id_to_delete INT;
    DECLARE cur CURSOR FOR
        SELECT id
        FROM rejestracja
        WHERE TIMESTAMPDIFF(MINUTE, czas, NOW()) >= 60
        ORDER BY czas ASC;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

    OPEN cur;

    read_loop: LOOP
```

```

        FETCH cur INTO id_to_delete;
        IF done THEN
            LEAVE read_loop;
        END IF;

        DELETE FROM rejestracja WHERE id = id_to_delete;
    END LOOP;

    CLOSE cur;

END//

```

Usuwanie zawartości tabeli blokada2 po godzinie gdzie przetrzymywane są ip od których ktoś próbował się zarejestrować bądź zmieniać hasło:

```

--usuwanie zawartości z zablokowaną możliwością wysyłania emaila do hasła i rejestracji
CREATE EVENT IF NOT EXISTS delete_expired_records_event3
ON SCHEDULE EVERY 1 MINUTE -- Sprawdzanie co minutę, czy istnieją rekordy do usunięcia
DO
BEGIN
    DECLARE done BOOLEAN DEFAULT FALSE;
    DECLARE id_to_delete INT;
    DECLARE cur CURSOR FOR
        SELECT id
        FROM blokada2
        WHERE TIMEDIFF(MINUTE, czas, NOW()) >= 60
        ORDER BY czas ASC;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

    OPEN cur;

    read_loop: LOOP
        FETCH cur INTO id_to_delete;
        IF done THEN
            LEAVE read_loop;
        END IF;

        DELETE FROM blokada2 WHERE id = id_to_delete;
    END LOOP;

    CLOSE cur;

END//

```

Wszystko znajduje się w baza.sql, events.sql i użytkownik.sql

2.Strona

index.php

Opis: Główna strona logowania, pierwszy punkt kontaktu dla użytkowników systemu.

Funkcje:

Logowanie użytkowników przez formularz przekazujący dane do skryptu/login.php.

Linki do stron przypomnij.php (reset hasła) i rejestracja.php (rejestracja nowych użytkowników).

Obsługa zmiany języka na ukraiński i przełączanie między motywem ciemnym a jasnym.

Technologie:

HTML dla struktury strony.

CSS z logowanie.css dla stylów.

PHP dla logiki backendowej (obsługa formularza).

JavaScript dla dynamicznych elementów, takich jak zmiana motywu.

```
<!--
  strona index.php
  strona startowa gdzie użytkownik ma możliwość zalogowania się na swoje konto
  lub przejście na stronę przypomnij.php jeżeli nie pamięta hasła
  lub przejście na stronę rejestracja.php jeżeli chce się zarejestrować
-->
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Logowanie</title>
  <link rel="stylesheet" href="style/logowanie.css">
</head>
<body>
  <!-- guzik od zmiany motywu z jasnego na ciemny i viceversa-->
  <button id="theme-toggle">Zmień motyw</button>
  <!-- guzik od zmiany języka z polskiego na ukraiński -->
  <form method="post">
    <button id="language-toggle" type="submit" name="jezyk">українська</button>
  </form>
  <?php
  // Obsługa przekierowania na inny język
  if (isset($_POST['jezyk'])) {
    header("location: ukr/index.php");
    exit;
  }
  ?>
```

Opis:

Rozpoczynamy od deklaracji typu dokumentu (<!DOCTYPE html>), który mówi przeglądarce, jak interpretować zawartość strony. <html lang="pl"> określa, że zawartość strony jest w języku polskim. W sekcji <head> znajdują się metadane, takie jak kodowanie znaków, viewport dla odpowiedniego wyświetlania na urządzeniach mobilnych, tytuł strony oraz link do arkusza stylów. <button id="theme-toggle">Zmień motyw</button> to przycisk, który po kliknięciu pozwala zmienić motyw strony. Wymaga dodatkowego obsługiwania w JavaScript, który nie został tutaj zawarty ponieważ jest on w późniejszej części tego pliku. Formularz do zmiany języka zawiera przycisk, który wysyła

żądanie POST z wartością język, obsługiwane przez PHP, które przekierowuje użytkownika na wersję strony w języku ukraińskim.

```
<script>
    // zmiana motywu
    const themeToggle = document.getElementById('theme-toggle');
    // Sprawdź, czy użytkownik ma zapisany preferowany motyw
    let currentTheme = localStorage.getItem('theme');
    if (!currentTheme) {
        // Jeśli nie ma zapisanego motywu, ustaw domyślnie motyw jasny
        currentTheme = 'light';
        localStorage.setItem('theme', currentTheme);
    }
    document.documentElement.setAttribute('data-theme', currentTheme);
    // Obsługa kliknięcia przycisku zmiany motywu
    themeToggle.addEventListener('click', () => {
        const currentTheme = document.documentElement.getAttribute('data-theme');
        const newTheme = currentTheme === 'light' ? 'dark' : 'light';
        document.documentElement.setAttribute('data-theme', newTheme);
        localStorage.setItem('theme', newTheme);
    });
</script>
<!-- formularz logowania -->
<div class="login-container">
    <h2>Zaloguj się</h2>
    <!-- skrypty/login.php -->
    <form action="skrypty/login.php" method="POST">
        <!-- email -->
        <div class="form-group">
            <input type="email" name="email" placeholder="Adres email" required>
        </div>
        <!-- hasło -->
        <div class="form-group">
            <input type="password" name="haslo" placeholder="Hasło" required>
        </div>
        <!-- przycisk od logowania -->
        <div class="form-group">
            <button type="submit" class="btn-login">Zaloguj się</button>
        </div>
    </form>
    <!-- przekierowanie jeżeli użytkownik nie pamięta hasła -->
    <div class="forgot-password-link">
        <span>Zapomniałeś hasła? <a href="przypomnij.php">Zresetuj je</a></span>
    </div>
    <!-- przekierowanie jeżeli użytkownik chce założyć konto -->
    <div class="register-link">
        <span>Nie masz jeszcze konta? <a href="rejestracja.php">Zarejestruj
się</a></span>
    </div>
    <?php
        // obsługa błędów
        // jeżeli dane są niepoprawne
        if(isset($_GET['error']) && $_GET['error'] == 1) {
            echo "<p class='error-message'>Nieprawidłowy email lub hasło</p>";
        }
        // jeżeli użytkownik został zablokowany z powodu za dużej liczby prób zalogowania
się
        if(isset($_GET['error']) && $_GET['error'] == 2) {
```

```

        echo "<p class='error-message'>Za dużo prób logowania<br>Spróbuj ponownie za 20
minut</p>";
    }
    // jeżeli użytkownik się zarejestrował lub zmienił hasło
    if(isset($_GET['login']) && $_GET['login'] == 1) {
        echo "<p class='good-message'>Możesz się już zalogować</p>";
    }
    ?>
</div>
</body>
</html>

```

Opis:

Fragment <script> obsługuje logikę zmiany motywu strony korzystając z localStorage do przechowywania preferencji użytkownika. Formularz logowania przesyła dane do skryptu/login.php i zawiera pola na email oraz hasło. Linki do resetowania hasła oraz rejestracji umożliwiają łatwe nawigowanie do odpowiednich stron. Obsługa błędów logowania jest realizowana poprzez parametry przekazywane metodą GET, które są wyświetlane jako komunikaty na stronie.

Przypomnij.php

Opis: Strona przeznaczona do użytkowników, którzy zapomnieli hasła.

Funkcje:

Formularz przekazujący dane do skryptu/przypomnij.php.

Obsługa zmiany języka na ukraiński i przełączanie między motywem ciemnym a jasnym.

Technologie:

HTML dla struktury strony.

CSS z logowanie.css dla stylów.

PHP dla logiki backendowej (obsługa formularza).

JavaScript dla dynamicznych elementów, takich jak zmiana motywu.

```

<!--
    strona przypomnij.php
    strona gdzie użytkownik ma możliwość zresetowania swojego hasła
-->
<!DOCTYPE html>
<html lang="pl">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Przypomnienie</title>
    <link rel="stylesheet" href="style/logowanie.css">
</head>
<body>
    <!-- guzik od zmiany motywu z jasnego na ciemny i viceversa-->

```

```

<button id="theme-toggle">Zmień motyw</button>
<!-- guzik od zmiany języka z polskiego na ukraiński -->
<form method="post">
  <button id="language-toggle" type="submit" name="jezyk">українська</button>
</form>
<!-- guzik X który cofa do index.php -->
<form method="post">
  <button id="exit" type="submit" name="exit">X</button>
</form>
<?php
// Obsługa przekierowania na inny język
if (isset($_POST['jezyk'])) {
  header("location: ukr/przypomnij.php");
  exit;
}
// Obsługa cofnięcia się
if (isset($_POST['exit'])) {
  header("location: index.php");
  exit;
}
?>

```

Opis:

Podobnie jak w index.php, strona rozpoczyna się od standardowego nagłówka HTML, włączając deklaracje metadanych oraz link do arkusza stylów. Trzy przyciski na górze strony umożliwiają zmianę motywu, zmianę języka oraz powrót do strony głównej (index.php). Obsługa przycisku zmiany języka i powrotu do strony głównej jest zaimplementowana za pomocą PHP, gdzie sprawdzane jest, czy odpowiednie wartości zostały przesłane metodą POST.

```

<script>
  // zmiana motywu
  const themeToggle = document.getElementById('theme-toggle');
  // Sprawdź, czy użytkownik ma zapisany preferowany motyw
  let currentTheme = localStorage.getItem('theme');
  if (!currentTheme) {
    // Jeśli nie ma zapisanego motywu, ustaw domyślnie motyw jasny
    currentTheme = 'light';
    localStorage.setItem('theme', currentTheme);
  }
  document.documentElement.setAttribute('data-theme', currentTheme);
  // Obsługa kliknięcia przycisku zmiany motywu
  themeToggle.addEventListener('click', () => {
    const currentTheme = document.documentElement.getAttribute('data-theme');
    const newTheme = currentTheme === 'light' ? 'dark' : 'light';
    document.documentElement.setAttribute('data-theme', newTheme);
    localStorage.setItem('theme', newTheme);
  });
</script>
<!-- formularz zmiany hasła -->
<div class="login-container">
  <h2>Nie pamiętam hasła</h2>
  <!-- przypomnij.php -->
  <form action="skrypty/przypomnij.php" method="POST">
    <!-- email -->
    <div class="form-group">
      <input type="email" name="email" placeholder="Adres email" required>
    </div>
  </form>
</div>

```

```

        <!-- przycisk który aktywuje resetowanie hasła -->
        <div class="form-group">
            <button type="submit" class="btn-login">Zresetuj hasło</button>
        </div>
    <?php
        // obsługa błędów
        // jeżeli email nie istnieje
        if(isset($_GET['error']) && $_GET['error'] == 1) {
            echo "<p class='error-message'>Podany email nie istnieje</p>";
        }
        // jeżeli już wysłano wiadomość w ciągu jednej godziny
        if(isset($_GET['error']) && $_GET['error'] == 2) {
            echo "<p class='error-message'>Za dużo wysłanych wiadomości<br>Spróbuj ponownie za godzinę</p>";
        }
    ?>
    </form>
</div>
</body>
</html>

```

Opis:

Skrypt JavaScript odpowiada za zachowanie ustawień motywu strony. Formularz umożliwia użytkownikom wprowadzenie adresu email, na który zostanie wysłana instrukcja resetowania hasła. W formularzu znajduje się również przycisk do wysyłania formularza, a PHP obsługuje ewentualne błędy, które mogą zostać wyświetlone użytkownikowi (np. email nie istnieje, za dużo prób resetowania hasła).

rejestracja.php

Opis: Strona przeznaczona do użytkowników, którzy pragną się zarejestrować.

Funkcje:

Formularz przekazujący dane do skryptu/rejestracja.php.

Obsługa zmiany języka na ukraiński i przełączanie między motywem ciemnym a jasnym.

Technologie:

HTML dla struktury strony.

CSS z logowanie.css dla stylów.

PHP dla logiki backendowej (obsługa formularza).

JavaScript dla dynamicznych elementów, takich jak zmiana motywu.

```

<!--
    strona rejestracja.php
    strona gdzie użytkownik ma możliwość zarejestrowania się
-->
<!DOCTYPE html>

```

```

<html lang="pl">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Rejestracja</title>
  <link rel="stylesheet" href="style/logowanie.css">
</head>
<body>
  <!-- guzik od zmiany motywu z jasnego na ciemny i viceversa-->
  <button id="theme-toggle">Zmień motyw</button>
  <!-- guzik od zmiany języka z polskiego na ukraiński -->
  <form method="post">
    <button id="language-toggle" type="submit" name="jezyk">українська</button>
  </form>
  <!-- guzik X który cofa do index.php -->
  <form method="post">
    <button id="exit" type="submit" name="exit">X</button>
  </form>
  <?php
  // Obsługa przekierowania na inny język
  if (isset($_POST['jezyk'])) {
    header("location: ukr/rejestracja.php");
    exit;
  }
  // Obsługa cofnięcia się
  if (isset($_POST['exit'])) {
    header("location: index.php");
    exit;
  }
  ?>

```

Opis:

Rozpoczyna od standardowej sekcji nagłówka HTML z odpowiednimi metadanymi, tytułem strony oraz linkiem do arkusza stylów CSS. Przyciski umożliwiają zmianę motywu, zmianę języka oraz powrót do strony głównej, podobnie jak w innych stronach. Kod PHP obsługuje przekierowania w zależności od akcji użytkownika (zmiana języka lub powrót do strony głównej).

```

<script>
  // zmiana motywu
  const themeToggle = document.getElementById('theme-toggle');
  // Sprawdź, czy użytkownik ma zapisany preferowany motyw
  let currentTheme = localStorage.getItem('theme');
  if (!currentTheme) {
    // Jeśli nie ma zapisanego motywu, ustaw domyślnie motyw jasny
    currentTheme = 'light';
    localStorage.setItem('theme', currentTheme);
  }
  document.documentElement.setAttribute('data-theme', currentTheme);
  // Obsługa kliknięcia przycisku zmiany motywu
  themeToggle.addEventListener('click', () => {
    const currentTheme = document.documentElement.getAttribute('data-theme');
    const newTheme = currentTheme === 'light' ? 'dark' : 'light';
    document.documentElement.setAttribute('data-theme', newTheme);
    localStorage.setItem('theme', newTheme);
  });
</script>
<!-- formularz rejestracji -->

```

```

<div class="login-container">
  <h2>Zarejestruj się</h2>
  <!-- rejestracja.php -->
  <form action="skrypty/rejestracja.php" method="POST"> <!-- Wskazujemy skrypt
obsługujący formularz -->
    <!-- email -->
    <div class="form-group">
      <input type="email" name="email" placeholder="Adres email" required>
    </div>
    <!-- przycisk który aktywuje rejestrację -->
    <div class="form-group">
      <button type="submit" class="btn-login">Zarejestruj się</button>
    </div>
  <?php
    // obsługa błędów
    // jeżeli email istnieje
    if(isset($_GET['error']) && $_GET['error'] == 1) {
      echo "<p class='error-message'>Podany email już istnieje</p>";
    }
    // jeżeli już wysłano wiadomość w ciągu jednej godziny
    if(isset($_GET['error']) && $_GET['error'] == 2) {
      echo "<p class='error-message'>Za dużo wysłanych wiadomości<br>Spróbuj
ponownie za godzinę</p>";
    }
  ?>
</form>
</div>
</body>
</html>

```

Opis:

JavaScript obsługuje logikę zmiany motywu strony. Formularz rejestracji zawiera pole na adres email oraz przycisk do przesyłania formularza do skryptu skrypty/rejestracja.php, który obsłuży dalszą logikę rejestracji. Komunikaty o błędach informują użytkownika o problemach, takich jak już istniejący email lub zbyt wiele prób wysłania formularza w krótkim czasie. Każdy z tych elementów odgrywa ważną rolę w zapewnieniu płynnej i bezpiecznej rejestracji użytkowników. Czy chcesz kontynuować analizę następnych plików z tej serii, tak aby przejść przez cały proces logiki aplikacji?

haslo.php

Opis: Strona przeznaczona tylko dla osób, które skorzystały z linka wysłanego po wypełnieniu formularza w rejestracja.php bądź przypomnij.php.

Funkcje:

Formularz przekazuje dane do zapytań SQL, a dane następnie wstawiane są do bazy danych

Obsługa zmiany języka na ukraiński i przełączanie między motywem ciemnym a jasnym.

Technologie:

HTML dla struktury strony.

CSS z logowanie.css dla stylów.

PHP dla logiki backendowej (obsługa formularza).

JavaScript dla dynamicznych elementów, takich jak zmiana motywu.

```
<!--
    strona haslo.php
    strona gdzie użytkownik ma możliwość ustawienia hasła jeżeli go zapomniał lub jeżeli
dopiero się zarejestrował
    może się na nią dostać tylko z linka który przychodzi do użytkownika poprzez wiadomość
email
-->
<?php
// Sprawdzenie, czy kod został przekazany jako parametr URL
if(isset($_GET['kod'])) {
    // Pobranie kodu z parametru URL
    $kod = $_GET['kod'];

    // Połączenie z bazą danych
    include('skrypty/baza.php');
    $polaczenie = mysqli_connect($host, $uzytkownik_bd, $haslo_bd, $bd);

    // Sprawdzenie, czy udało się połączyć z bazą danych
    if (!$polaczenie) {
        die("Błąd połączenia z bazą danych: " . mysqli_connect_error());
    }

    // Zabezpieczenie przed atakami SQL injection
    $kod = mysqli_real_escape_string($polaczenie, $kod);

    // Zapytanie do bazy danych w celu sprawdzenia, czy kod istnieje w tabeli rejestracja
    $query = "SELECT * FROM rejestracja WHERE kod='$kod'";
    $result = mysqli_query($polaczenie, $query);

    // Sprawdzenie, czy kod został znaleziony w bazie danych
    if(mysqli_num_rows($result) == 1) {
        // Użytkownik ma poprawny kod - wyświetlenie formularza zmiany hasła
    }
}

?>

<!DOCTYPE html>
<html lang="pl">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Ustawienie hasła</title>
    <link rel="stylesheet" href="style/logowanie.css">
</head>
<body>
    <!-- guzik od zmiany motywu z jasnego na ciemny i viceversa-->
    <button id="theme-toggle">Zmień motyw</button>
    <!-- guzik od zmiany języka z polskiego na ukraiński -->
    <form method="post">
        <button id="language-toggle" type="submit" name="jezyk">українська</button>
    </form>
    <?php
    // Obsługa przekierowania na inny język
    if (isset($_POST['jezyk'])) {
```

```

        header("location: ukr/haslo.php?kod=$kod");
        exit;
    }
    ?>
<script>
    const themeToggle = document.getElementById('theme-toggle');
    // Sprawdź, czy użytkownik ma zapisany preferowany motyw
    let currentTheme = localStorage.getItem('theme');
    if (!currentTheme) {
        // Jeśli nie ma zapisanego motywu, ustaw domyślnie motyw jasny
        currentTheme = 'light';
        localStorage.setItem('theme', currentTheme);
    }
    document.documentElement.setAttribute('data-theme', currentTheme);
    // Obsługa kliknięcia przycisku zmiany motywu
    themeToggle.addEventListener('click', () => {
        const currentTheme = document.documentElement.getAttribute('data-theme');
        const newTheme = currentTheme === 'light' ? 'dark' : 'light';
        document.documentElement.setAttribute('data-theme', newTheme);
        localStorage.setItem('theme', newTheme);
    });
</script>
<!-- formularz ustawienia hasła -->
<div class="login-container">
    <h2>Podaj hasło</h2>
    <!-- haslo.php -->
    <form action="skrypty/haslo.php?kod=<?php echo $kod; ?>" method="POST"> <!--
Wskazujemy skrypt obsługujący formularz -->
        <!-- hasło -->
        <div class="form-group">
            <input type="password" name="haslo" id="haslo" placeholder="hasło"
required>
        </div>
        <!-- przycisk który aktywuje ustawienie hasła -->
        <div class="form-group">
            <button type="submit" class="btn-login">Zatwierdź</button>
        </div>
    </form>
<script>
    // Walidacja hasła po stronie klienta
    document.addEventListener('DOMContentLoaded', function() {
        // Pobranie referencji do pola hasła
        const passwordInput = document.getElementById('haslo');

        // Funkcja sprawdzająca poprawność hasła
        function validatePassword() {
            const password = passwordInput.value;

            // Sprawdzenie, czy hasło zawiera minimum 8 znaków
            if (password.length < 8) {
                passwordInput.setCustomValidity('Hasło musi zawierać co najmniej 8
znaków.');
```



```

        passwordInput.setCustomValidity('Hasło musi zawierać co najmniej
jedną dużą literę.');
```

```

        return;
    }

    // Sprawdzenie, czy hasło zawiera co najmniej jedną cyfrę
    if (!/\d/.test(password)) {
        passwordInput.setCustomValidity('Hasło musi zawierać co najmniej
jedną cyfrę.');
```

```

        return;
    }

    // Sprawdzenie, czy hasło zawiera co najmniej jeden znak specjalny
    if (!/[!@%*?&]/.test(password)) {
        passwordInput.setCustomValidity('Hasło musi zawierać co najmniej
jeden znak specjalny.');
```

```

        return;
    }

    // Hasło spełnia wszystkie kryteria - brak komunikatu o błędzie
    passwordInput.setCustomValidity('');
}

// Nasłuchiwanie na zmiany w polu hasła i wywołanie funkcji walidacji
passwordInput.addEventListener('input', validatePassword);
});
</script>
</div>
</body>
</html>

<?php
    } else {
        // Kod nie został znaleziony w bazie danych - przekierowanie do strony głównej
        header("location: index.php");
        exit;
    }

    // Zamknięcie połączenia z bazą danych
    mysqli_close($polaczenie);
} else {
    // Jeśli kod nie został przekazany jako parametr URL, przekierowanie do strony głównej
    header("location: index.php");
    exit;
}
?>

```

Opis:

Bezpieczeństwo i walidacja: Kod przekazywany jest przez parametr URL i jest sprawdzany przed wykonaniem jakichkolwiek operacji na bazie danych. Użyto funkcji `mysqli_real_escape_string` do ochrony przed SQL Injection. Formularz ustawienia hasła: Wyświetlany tylko, jeśli kod przesłany w URL jest prawidłowy i istnieje w tabeli rejestracja. To zapewnia dodatkową warstwę bezpieczeństwa, upewniając się, że tylko osoby z ważnym kodem mogą zresetować hasło. Przekierowania: Jeśli kod jest nieprawidłowy lub nie został przekazany, użytkownik jest automatycznie przekierowywany na stronę główną, co zapobiega przypadkowemu dostępowi do formularza resetowania hasła. Interfejs użytkownika: Formularz na tej stronie pozwala użytkownikowi na wprowadzenie nowego hasła, które

musi spełniać określone kryteria bezpieczeństwa (te kryteria są określone w dostarczonym kodzie).
Obsługa języków: Tak jak w innych częściach aplikacji, użytkownik może zmienić język interfejsu.
Przekierowania uwzględniają wybrany język, co pozwala na łatwe i spójne zarządzanie lokalizacją.

user.php

Opis: Strona użytkownika, na której może on przechowywać swoje pliki.

Funkcje:

Możliwość podglądu swojego maksymalnego miejsca oraz tego, które aktualnie użytkownik zajmuje.

Możliwość wrzucania, przechowywania, pobierania, usuwania i udostępniania plików.

Możliwość filtrowania konkretnych plików.

Obsługa zmiany języka na ukraiński i przełączanie między motywem ciemnym a jasnym.

Technologie:

HTML dla struktury strony.

CSS z uzytkownik.css dla stylów.

PHP dla logiki backendowej (obsługa formularza).

JavaScript dla dynamicznych elementów, takich jak zmiana motywu.

```
<!--
    strona user.php
    strona przeznaczona dla użytkowników z której mają możliwość wrzucania plików do chmury
    oraz mają parę opcji robienia czegoś z nimi (pobranie, usunięcie, udostępnienie)
-->
<?php
session_start();

// Sprawdzenie czy użytkownik jest zalogowany
if (!isset($_SESSION['zalogowany']) || $_SESSION['zalogowany'] !== true) {
    // Jeżeli nie, przekieruj na stronę logowania
    header("location: index.php");
    exit;
}

// Sprawdzenie czy użytkownik ma uprawnienia użytkownika
if (!isset($_SESSION['typ_uzytkownika']) || $_SESSION['typ_uzytkownika'] !== 'user') {
    // Jeżeli nie, przekieruj na stronę główną
    header("location: index.php");
    exit;
}

// Obsługa wylogowania
if (isset($_POST['wyloguj'])) {
    // Usunięcie zmiennych sesji
```

```

unset($_SESSION['zalogowany']);
unset($_SESSION['typ_uzytkownika']);
// Przekierowanie na stronę logowania
header("location: index.php");
exit;
}
// Obsługa przekierowania z polskiego na ukraiński
if (isset($_POST['jezyk'])) {
    // Przekierowanie na stronę logowania
    header("location: ukr/user.php");
    exit;
}
?>

```

Opis:

Zarządzanie sesją: Plik rozpoczyna się od sprawdzenia, czy użytkownik jest zalogowany i ma odpowiednie uprawnienia (user). Jest to kluczowe dla zapewnienia, że tylko zautoryzowane osoby mają dostęp do tej strony. Wylogowanie: Użytkownik może wylogować się z systemu, co skutkuje usunięciem danych sesji i przekierowaniem na stronę logowania. Zmiana języka: Użytkownicy mogą zmienić język interfejsu, co jest obsługiwane przez przekierowanie na odpowiednią wersję językową strony.

```

<!DOCTYPE html>
<html lang="pl">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Strona użytkownika</title>
    <link rel="stylesheet" href="style/uzytkownik.css">
</head>
<body>
    <div class="lewy">
        <?php
            // Połączenie z bazą danych
            require_once('skrypty/baza.php');
            $conn = new mysqli($host, $uzytkownik_bd, $haslo_bd, $bd);

            // Sprawdzenie połączenia
            if ($conn->connect_error) {
                die("Błąd połączenia z bazą danych: " . $conn->connect_error);
            }
            $user_email = $_SESSION['email'];
            echo $user_email;
            echo "<h1>Witaj na stronie użytkownika</h1>";

            // Pobranie miejsca użytkownika
            $query = "SELECT miejsce FROM uzytkownicy WHERE email = ?";
            $stmt = $conn->prepare($query);
            $stmt->bind_param("s", $user_email);
            $stmt->execute();
            $result = $stmt->get_result();
            $miejsce_uzytkownika = $result->fetch_assoc()['miejsce']; // Pobranie miejsca
uzytkownika
            $stmt->close();

            // Pobranie sumy rozmiarów plików użytkownika

```

```

        $query_zajete_miejsce = "SELECT SUM(p.bajty) AS zajete_miejsce FROM uzytkownicy u
INNER JOIN pliki p ON u.id = p.wlasciciel WHERE u.email = ?";
        $stmt_zajete_miejsce = $conn->prepare($query_zajete_miejsce);
        $stmt_zajete_miejsce->bind_param("s", $user_email);
        $stmt_zajete_miejsce->execute();
        $result_zajete_miejsce = $stmt_zajete_miejsce->get_result();
        $stmt_zajete_miejsce->close();

        if ($result_zajete_miejsce->num_rows == 1) {
            $row_zajete_miejsce = $result_zajete_miejsce->fetch_assoc();
            $zajete_miejsce_bajty = $row_zajete_miejsce['zajete_miejsce']; // Suma zajętego
miejsca w bajtach

            // Konwersja bajtów na megabajty
            $zajete_miejsce_MB = round($zajete_miejsce_bajty / (1024 * 1024), 2); //
Zaokrąglamy do dwóch miejsc po przecinku

            // Wyświetlenie wyniku
            echo "<h3>Ilość miejsca na dysku:</h3>";
            echo "<p>" . $zajete_miejsce_MB . "MB / " . $miejsce_uzytkownika . "MB</p>";
        } else {
            echo "Błąd: Nie można znaleźć sumy zajętego miejsca użytkownika.";
        }
    }
}

<!-- Przycisk wylogowania -->
<form method="post">
    <button id="logout" type="submit" name="wyloguj">Wyloguj</button>
</form>

<!-- przycisk zmiany motywu -->
<button id="theme-toggle">Zmień motyw</button>
<form method="post">
    <button id="language-toggle" type="submit" name="jezyk">українська</button>
</form>
</div>

<div class="prawy">
    <!-- Formularz wyszukiwania plików -->
    <h2>Wyszukiwanie plików</h2>
    <form method="GET">
        <input type="text" name="frazza" placeholder="Wprowadź frazę">
        <button class='button' type="submit">Szukaj</button>
    </form><br>
    <!-- Formularz przesyłania pliku -->
    <form method="post" enctype="multipart/form-data" action="skrypty/dodaj_plik.php">
        <input type="file" name="file" id="fileInput" onchange="updateFileMessage()">
        <label for="fileInput" id="fileInputLabel">Wybierz plik</label>
        <span id="fileMessage">Nie wybrano pliku</span>
        <button class='button' type="submit" name="submit">Prześlij plik</button>
    </form>
    <?php
    // obsługa błędów
    // jeżeli plik nie może być znaleziony
    if(isset($_GET['error']) && $_GET['error'] == 1) {
        echo "<p class='error-message'>Błąd: Nie można znaleźć pliku dla tego
użytkownika.</p>";
    }
    // jeżeli nie można znaleźć użytkownika

```

```

if(isset($_GET['error']) && $_GET['error'] == 2) {
    echo "<p class='error-message'>Błąd: Nie można znaleźć użytkownika.</p>";
}
// jeżeli id pliku jest błędne
if(isset($_GET['error']) && $_GET['error'] == 3) {
    echo "<p class='error-message'>Błąd: Błąd identyfikatora pliku</p>";
}
// jeżeli nie uda się przesłać pliku
if(isset($_GET['error']) && $_GET['error'] == 4) {
    echo "<p class='error-message'>Błąd: Nie udało się przesłać pliku</p>";
}
//Wyświetlanie plików użytkownika
echo "<h2>Twoje pliki</h2>";

// Pobranie listy plików użytkownika z uwzględnieniem wyszukiwanej frazy
if(!empty($_GET['fraz'])) {
    $fraz = '%' . $_GET['fraz'] . '%';
    $query_files = "SELECT p.id, p.nazwa, p.kod
FROM pliki p
INNER JOIN uzytkownicy u ON p.wlasciciel = u.id
WHERE u.email = ? AND p.nazwa LIKE ?";
    $stmt_files = $conn->prepare($query_files);
    $stmt_files->bind_param("ss", $user_email, $fraz);
} else {
    // Pobranie wszystkich plików użytkownika, jeśli nie ma frazy wyszukiwania
    $query_files = "SELECT p.id, p.nazwa, p.kod
FROM pliki p
INNER JOIN uzytkownicy u ON p.wlasciciel = u.id
WHERE u.email = ?";
    $stmt_files = $conn->prepare($query_files);
    $stmt_files->bind_param("s", $user_email);
}

$stmt_files->execute();
$result_files = $stmt_files->get_result();
// jeżeli są pliki to je wyświetl razem z opcjami
if ($result_files->num_rows > 0) {
    echo '<ul>';
    // tworzenie listy plików
    while ($row_file = $result_files->fetch_assoc()) {
        // nazwa pliku
        echo '<li><h3>' . htmlspecialchars(urldecode($row_file['nazwa']));
        // możliwość pobrania pliku
        echo '</h3></li><li> <form method="post" action="skrypty/pobierz_plik.php"
style="display:inline-block;">
            <input type="hidden" name="plik_id" value="' . $row_file['id'] . '">
            <button class="button" type="submit" name="pobierz">Pobierz</button>
        </form>';
        // możliwość usunięcia pliku
        echo ' <form method="post" action="skrypty/usun_plik.php"
style="display:inline-block;">
            <input type="hidden" name="plik_id" value="' . $row_file['id'] . '">
            <button class="button" type="submit" name="usun">Usuń</button>
        </form>';

        // Udostępnianie pliku
        if(empty($row_file['kod'])) {

```

```

        echo ' <form method="post" action="skrypty/udostepnij_plik.php"
style="display:inline-block;">
            <input type="hidden" name="plik_id" value="' . $row_file['id'] .
'">
            <button class="button" type="submit"
name="udostepnij">Udostępnij</button>
            </form>';
    } else {
        // Wyświetlanie linku do udostępnionego pliku
        $link = $moja_strona . "skrypty/plik.php?kod=" . $row_file['kod'];
        // skopiowanie pliku
        echo ' <button class="button" type="submit"
onclick="copyToClipboard(\'' . $link . '\')">Kopiuż link</button> ';
        // jeżeli plik jest udostępniony to możliwość przestania udostępniania
        echo ' <form method="post" action="skrypty/przestan_udostepniac.php"
style="display:inline-block;">
            <input type="hidden" name="plik_id" value="' . $row_file['id'] . '">
            <button class="button" type="submit"
name="przestan_udostepniac">Przestań udostępniać</button>
            </form>';
    }

    echo '</li><li><br></li>';
}
echo '</ul>';
} else {
    // jeżeli użytkownik nie ma plików to komunikat brak plików
    echo '<p>Brak plików.</p>';
}
// Zamknięcie połączenia z bazą danych
$stmt_files->close();
$conn->close();
?>
</div>
<!-- Wyczyszczenie floatów bo bez tego wygląd strony się psuje-->
<div class="clearfix"></div>

<script>
// Funkcja do kopiowania linku do schowka
function copyToClipboard(text) {
    var dummy = document.createElement("textarea");
    document.body.appendChild(dummy);
    dummy.value = text;
    dummy.select();
    document.execCommand("copy");
    document.body.removeChild(dummy);
    alert("Skopiowano link do schowka: " + text);
}
// funkcja do zmiany motywu strony
const themeToggle = document.getElementById('theme-toggle');
// Sprawdź, czy użytkownik ma zapisany preferowany motyw
let currentTheme = localStorage.getItem('theme');
if (!currentTheme) {
    // Jeśli nie ma zapisanego motywu, ustaw domyślnie motyw jasny
    currentTheme = 'light';
    localStorage.setItem('theme', currentTheme);
}
document.documentElement.setAttribute('data-theme', currentTheme);

```

```

// Obsługa kliknięcia przycisku zmiany motywu
themeToggle.addEventListener('click', () => {
  const currentTheme = document.documentElement.getAttribute('data-theme');
  const newTheme = currentTheme === 'light' ? 'dark' : 'light';
  document.documentElement.setAttribute('data-theme', newTheme);
  localStorage.setItem('theme', newTheme);
});
// funkcja do modyfikacji podstawowego napisu "nie wybrano pliku" i "Wybrano
plik:..."
function updateFileMessage() {
  var fileInput = document.getElementById('fileInput');
  var fileMessage = document.getElementById('fileMessage');

  if (fileInput.files.length > 0) {
    fileMessage.textContent = 'Wybrano plik: ' + fileInput.files[0].name;
  } else {
    fileMessage.textContent = 'Nie wybrano pliku';
  }
}
</script>
</body>
</html>

```

Opis końcowy i funkcjonalność:

Wyszukiwanie plików: Użytkownik może wyszukać pliki wg frazy. Wyniki są filtrowane i wyświetlane.

Przesyłanie plików: Formularz pozwala na wysyłanie plików do serwera. Obsługa błędów informuje użytkownika o problemach z przesyłaniem.

Zarządzanie plikami: Użytkownik może pobierać, usuwać i udostępniać pliki. Udostępnione pliki otrzymują unikalny kod, który może być kopiowany do schowka.

Motywy i języki: Strona wspiera zmianę motywów oraz języków, co poprawia użyteczność dla międzynarodowych użytkowników.

Zarządzanie widocznością plików: Dostępne są opcje udostępniania i przestawiania udostępniania, co zwiększa kontrolę użytkownika nad jego danymi.

admin.php

Opis: Strona admina, na której może on kontrolować użytkowników.

Funkcje:

Możliwość podglądu statystyk takich jak maksymalne miejsce użytkowników czy też miejsce które aktualnie zajmują.

Możliwość usuwania użytkowników z wcześniejszym alertem.

Możliwość edycji miejsca użytkownika.

Obsługa zmiany języka na ukraiński i przełączanie między motywem ciemnym a jasnym.

Technologie:

HTML dla struktury strony.

CSS z uzytkownik.css dla stylów.

PHP dla logiki backendowej (obsługa formularza).

JavaScript dla dynamicznych elementów, takich jak zmiana motywu.

```
<!--
    strona admin.php
    strona przeznaczona dla administratora z której ma podgląd na statystyki oraz ma
    możliwość edycji użytkowników
-->
<?php
session_start();

// Sprawdzenie czy użytkownik jest zalogowany
if (!isset($_SESSION['zalogowany']) || $_SESSION['zalogowany'] !== true) {
    // Jeżeli nie, przekieruj na stronę logowania
    header("location: index.php");
    exit;
}

// Sprawdzenie czy użytkownik ma uprawnienia administratora
if (!isset($_SESSION['typ_uzytkownika']) || $_SESSION['typ_uzytkownika'] !== 'admin') {
    // Jeżeli nie, przekieruj na stronę główną
    header("location: index.php");
    exit;
}

// Obsługa wylogowania
if (isset($_POST['wyloguj'])) {
    // Usunięcie zmiennych sesji
    unset($_SESSION['zalogowany']);
    unset($_SESSION['typ_uzytkownika']);
    // Przekierowanie na stronę logowania
    header("location: index.php");
    exit;
}

// Obsługa przekierowania z polskiego na ukraiński
if (isset($_POST['jezyk'])) {
    header("location: ukr/admin.php");
    exit;
}

// Połączenie z bazą danych
require_once('skrypty/baza.php');
$conn = new mysqli($host, $uzytkownik_bd, $haslo_bd, $bd);

// Sprawdzenie połączenia
if ($conn->connect_error) {
    die("Błąd połączenia z bazą danych: " . $conn->connect_error);
}
?>
```

Opis:

Zarządzanie sesją: Strona rozpoczyna od sprawdzenia, czy użytkownik jest zalogowany i czy posiada uprawnienia administratora. To zapewnia, że tylko upoważnieni administratorzy mają dostęp do tej części systemu. Wylogowanie i zmiana języka: Administrator może się wylogować lub zmienić język interfejsu, podobnie jak na innych stronach.

```
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Strona administratora</title>
  <link rel="stylesheet" href="style/uzytkownik.css">
</head>
<body>
  <div class="lewy">
    <?php echo $_SESSION['email'] ?>
    <h1>Witaj na stronie administratora</h1>

    <!-- Przycisk wylogowania -->
    <form method="post">
      <button id="logout" type="submit" name="wyloguj">Wyloguj</button>
    </form>
    <!-- motyw -->
    <button id="theme-toggle">Zmień motyw</button>
    <!-- język -->
    <form method="post">
      <button id="language-toggle" type="submit" name="jezyk">українська</button>
    </form>
  </div>
  <div class="prawy">
    <?php
      // Pobranie sumy rozmiarów plików użytkownika
      $query_zajete_miejsce = "SELECT SUM(bajty) AS zajete_miejsce FROM pliki";
      $stmt_zajete_miejsce = $conn->prepare($query_zajete_miejsce);
      $stmt_zajete_miejsce->execute();
      $result_zajete_miejsce = $stmt_zajete_miejsce->get_result();
      $stmt_zajete_miejsce->close();
      if ($result_zajete_miejsce->num_rows == 1) {
        $row_zajete_miejsce = $result_zajete_miejsce->fetch_assoc();
        $zajete_miejsce_bajty = $row_zajete_miejsce['zajete_miejsce']; // Suma zajętego
miejsca w bajtach
        // Konwersja bajtów na megabajty
        $zajete_miejsce_MB = round($zajete_miejsce_bajty / (1024 * 1024), 2); //
Zaokrąglamy do dwóch miejsc po przecinku
        echo "<h3>Ilość zajętego miejsca miejsca na dysku:</h3>";
        echo "<p>" . $zajete_miejsce_MB . "MB</p>";
      } else {
        echo "Błąd: Nie można znaleźć sumy zajętego miejsca użytkownika.";
      }
    <?>
    <!-- lista użytkowników razem z ich emailem, zajęтым miejscem, ograniczeniem które
można zmienić, możliwością ich usunięcia -->
    <h3>Lista użytkowników:</h3>
    <table>
      <tr>
        <th>Email</th>
        <th>Zajęte miejsce (MB)</th>
```

```

        <th>Ograniczenie (MB)</th>
    </tr>
<?php
    // Pobranie listy użytkowników z informacjami
    $query_users = "SELECT u.email, IFNULL(round(SUM(p.bajty)/(1024*1024), 2), 0)
AS zajete_miejsce, u.miejsce FROM uzytkownicy u LEFT JOIN pliki p ON u.id=p.wlasciciel
WHERE u.admin=0 GROUP BY u.email ORDER BY zajete_miejsce DESC";
    $result_users = $conn->query($query_users);
    if ($result_users->num_rows > 0) {
        while ($row_user = $result_users->fetch_assoc()) {
            $email = $row_user['email'];
            $zajete_miejsce = $row_user['zajete_miejsce'];
            $miejsce = $row_user['miejsce'];
            echo "<tr>";
            echo "<td>$email</td>";
            echo "<td>$zajete_miejsce</td>";
            echo "<td><form method='post' action='skrypty/zmien_miejsce.php'><input
type='hidden' name='email' value='$email'><input class='button' type='number'
name='nowe_miejsce' value='$miejsce'><button class='button' type='submit'
name='zmien'>Zmień miejsce</button></form></td>";
            echo "<td><form method='post' action='skrypty/usun_uzytkownika.php'
onsubmit='return confirm(\"Czy na pewno chcesz usunąć użytkownika $email?\")'><input
type='hidden' name='email' value='$email'><button class='button' type='submit'
name='usun'>Usuń użytkownika</button></form></td>";
            echo "</tr>";
        }
    } else {
        echo "<tr><td colspan='3'>Brak użytkowników.</td></tr>";
    }
    ?>
</table>
</div>
<?php
$conn->close();
?>
<script>
    // zmiana motywu
    const themeToggle = document.getElementById('theme-toggle');
    // Sprawdź, czy użytkownik ma zapisany preferowany motyw
    let currentTheme = localStorage.getItem('theme');
    if (!currentTheme) {
        // Jeśli nie ma zapisanego motywu, ustaw domyślnie motyw jasny
        currentTheme = 'light';
        localStorage.setItem('theme', currentTheme);
    }
    document.documentElement.setAttribute('data-theme', currentTheme);
    // Obsługa kliknięcia przycisku zmiany motywu
    themeToggle.addEventListener('click', () => {
        const currentTheme = document.documentElement.getAttribute('data-theme');
        const newTheme = currentTheme === 'light' ? 'dark' : 'light';
        document.documentElement.setAttribute('data-theme', newTheme);
        localStorage.setItem('theme', newTheme);
    });
</script>
</body>
</html>

```

Opis końcowy i funkcjonalność:

Podgląd statystyk: Administrator może przeglądać całkowitą ilość miejsca zajmowanego przez wszystkie pliki przechowywane w systemie.

Zarządzanie użytkownikami: Administrator ma możliwość przeglądania listy wszystkich użytkowników, ich zajętego miejsca oraz dostępnego limitu. Może zmieniać limit miejsca użytkownika oraz usuwać użytkowników z systemu.

Formularze akcji: Dla każdego użytkownika dostępne są formularze umożliwiające edycję limitu miejsca oraz usunięcie konta, z dodatkową warstwą potwierdzenia akcji usuwania, co zabezpiecza przed przypadkowym usunięciem.

Interaktywne Elementy: Przyciski tematyczne i językowe: Admin ma możliwość zmiany motywu i języka strony dla lepszego dostosowania do swoich preferencji. Zarządzanie miejsce użytkowników: Admin może dynamicznie zarządzać przydziałem przestrzeni dyskowej dla poszczególnych użytkowników, co jest szczególnie przydatne w środowiskach z ograniczonymi zasobami. Bezpieczeństwo: Formularze akcji są zabezpieczone poprzez dialogi potwierdzające, co minimalizuje ryzyko błędów ludzkich przy zarządzaniu użytkownikami.

3.CSS

Opis:

Arkusze stylów CSS definiujące wygląd strony logowania oraz panelu użytkownika.

Określają styl formularzy, przycisków, linków, a także motywy ciemne i jasne.

uzytkownik.css przeznaczony jest dla admin.php i user.php

logowanie.css przeznaczony jest dla pozostałych stron znajdujących się w sekcji Strony

logowanie.css

```
/* przycisk od zmiany języka */
#language-toggle {
    position: fixed;
    top: 60px; /* Zmieniona wartość top, aby przycisk był umieszczony poniżej */
    right: 20px;
    z-index: 9999;
    background-color: #fff;
    border: 1px solid #000;
    padding: 10px 20px;
    cursor: pointer;
}

/* przycisk od zmiany motywu */
#theme-toggle {
    position: fixed;
    top: 20px;
    right: 20px;
    z-index: 9999;
    background-color: #fff;
    border: 1px solid #000;
    padding: 10px 20px;
    cursor: pointer;
}
```

```

}
/* przycisk od wyjścia ze strony */
#exit {
    position: fixed;
    top: 20px;
    left: 20px;
    z-index: 9999;
    background-color: #fff;
    border: 1px solid #000;
    padding: 10px 20px;
    cursor: pointer;
}
/* Motyw jasny */
[data-theme="light"] {
    --background-color: #f4f4f4;
    --text-color: #000;
    --border-color: #000;
}
/* Motyw ciemny */
[data-theme="dark"] {
    --background-color: #040404;
    --text-color: #fff;
    --border-color: #fff;
}
/* wygląd body */
body {
    font-family: Arial, sans-serif;
    background-color: var(--background-color);
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}
/* wygląd okienka */
.login-container {
    background-color: #fff;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
    text-align: center;
}
/* h2 w okienku */
.login-container h2 {
    margin-bottom: 20px;
}
.form-group {
    margin-bottom: 20px;
}
.form-group input {
    width: 100%;
    padding: 10px;
    border-radius: 4px;
    border: 1px solid #ccc;
    box-sizing: border-box;
}

```

```

/* przycisk logowania, zmiany hasła itp. */
.btn-login {
    background-color: #4caf50;
    color: #fff;
    border: none;
    padding: 10px 20px;
    border-radius: 4px;
    cursor: pointer;
}
/* podczas najechania na przycisk */
.btn-login:hover {
    background-color: #45a049;
}
/* link rejestracji */
.register-link {
    margin-top: 15px;
}
/* a href linku */
.register-link a {
    color: #007bff;
    text-decoration: none;
}
/* podczas najechania na link */
.register-link a:hover {
    text-decoration: underline;
}
/* link zapomnienie hasła */
.forgot-password-link {
    margin-top: 15px;
}
/* a href linku */
.forgot-password-link a {
    color: #007bff;
    text-decoration: none;
}
/* podczas najechania na link */
.forgot-password-link a:hover {
    text-decoration: underline;
}
/* error */
.error-message {
    color: red;
}
/* dobre wiadomości, potwierdzenia */
.good-message {
    color: rgb(0, 255, 0);
}

```

uzytkownik.css

```

/* wygląd przycisku od zmiany języka */
#language-toggle {
    position: fixed;
    top: 60px; /* Zmieniona wartość top, aby przycisk był umieszczony poniżej */
    right: 20px;
    z-index: 9999;
    background-color: #fff;
}

```

```

    border: 1px solid #000;
    padding: 10px 20px;
    cursor: pointer;
}
/* wygląd przycisku od zmiany wyglądu */
#theme-toggle {
    position: fixed;
    top: 20px;
    right: 20px;
    z-index: 9999;
    background-color: #fff;
    border: 1px solid #000;
    padding: 10px 20px;
    cursor: pointer;
}
/* wygląd przycisku do wylogowania */
#logout {
    position: fixed;
    bottom: 20px;
    left: 20px;
    z-index: 9999;
    background-color: #fff;
    border: 1px solid #000;
    padding: 10px 20px;
    cursor: pointer;
}
/* wygląd przycisku */
.button {
    z-index: 9999;
    background-color: #fff;
    border: 1px solid #000;
    padding: 10px 20px;
    cursor: pointer;
}
/* ukrycie defaultowego przycisku "dodaj plik" */
#fileInput {
    display: none; /* ukryj domyślny przycisk */
}
/* przycisk własny */
#fileInputLabel {
    z-index: 9999;
    background-color: #fff;
    border: 1px solid #000;
    padding: 10px 20px;
    cursor: pointer;
    color: black;
}
/* Motyw jasny */
[data-theme="light"] {
    --background-color: #f4f4f4;
    --text-color: #000;
    --border-color: #000;
}
/* Motyw ciemny */
[data-theme="dark"] {
    --background-color: #000;
    --text-color: #fff;
    --border-color: #fff;
}

```

```

}
/* wygląd body */
body {
    background-color: var(--background-color);
    color: var(--text-color);
}
/* lewa część strony */
.lewy {
    position: fixed;
    top: 0;
    left: 0;
    width: 30%;
    height: 100%;
    overflow-y: auto;
    padding: 20px;
    box-sizing: border-box;
}
/* prawa część strony */
.prawy {
    margin-left: 30%; /* Ustawienie marginesu, aby uniknąć zasłaniania przez lewy div */
    padding: 20px;
    box-sizing: border-box;
}
/* Wyczyszczenie floatów bo bez tego strona źle wygląda*/
.clearfix::after {
    content: "";
    clear: both;
    display: table;
}
/* wygląd errorów */
.error-message {
    color: red;
}
/* wygląd dobrych wiadomości */
.good-message {
    color: rgb(0, 255, 0);
}
/* Styl dla tabeli */
table {
    margin: 0 auto;
    text-align: center;
}

/* Styl dla nagłówków tabeli */
th {
    text-align: center;
}
/* Styl dla komórek tabeli */
table td {
    text-align: center;
    border: 1px solid var(--border-color);; /* Dodajemy obramowanie 1px czarnym kolorem */
}
/* styl dla ul */
ul {
    list-style-type: none;
}

```

4.Skrypty

W sekcji skrypty znajdują się skrypty znajdujące się w folderze skrypty.

Oprócz skryptów znajdujących się poniżej, jest też biblioteka PHPMailer 6.9.1

baza.php

Plik konfiguracyjny zawierający dane do połączenia z bazą danych oraz bazowy URL strony.

```
<!--
    baza.php
    skrypt który ma za zadanie przetrzymywać dane dotyczące łączenia się z bazą danych oraz
    ścieżka strony
-->
<?php
$host=""; //ip gdzie znajduje się baza danych
$uzytkownik_bd=""; //nazwa użytkownika do bazy danych
$haslo_bd=""; //hasło do bazy danych
$bd="projekt_chmura"; //nazwa bazy danych
$moja_strona="https://jakubmartynski.pl/projekt/"; //ścieżka strony gdzie znajduje się
index.php(strona startowa)
?>
```

Opis:

Centralizacja konfiguracji: Plik baza.php centralizuje informacje o połączeniu z bazą danych, co ułatwia zarządzanie zmianami w konfiguracji i ponowne użycie kodu.

Bezpieczeństwo: Dane takie jak nazwa użytkownika i hasło do bazy danych są przechowywane w jednym miejscu, co powinno być zabezpieczone odpowiednimi uprawnieniami dostępu do pliku.

dodaj_plik.php

Obsługuje dodawanie plików do systemu.

```
<!--
    dodaj_plik.php
    skrypt odpowiedzialny za dodanie pliku do bazy danych
-->
<?php
session_start();
require_once('baza.php');
// sprawdzenie czy dane zostały wprowadzone prawidłowo
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_FILES["file"]) &&
isset($_SESSION['email'])) {
    $file_name = $_FILES["file"]["name"];
    $file_temp = $_FILES["file"]["tmp_name"];
    $file_size = $_FILES["file"]["size"];
    $user_email = $_SESSION['email'];
    // sprawdzenie czy użytkownik wrzucił jakikolwiek plik
    if($file_name == NULL){
        header("Location: ../user.php");
    }
    // Połączenie z bazą danych
```



```

$conn = new mysqli($host, $uzytkownik_bd, $haslo_bd, $bd);
// Sprawdzanie połączenia
if ($conn->connect_error) {
    die("Błąd połączenia z bazą danych: " . $conn->connect_error);
}

// Pobranie sumy rozmiarów plików użytkownika
$query_zajete_miejsce = "SELECT SUM(p.bajty) AS zajete_miejsce FROM uzytkownicy u INNER
JOIN pliki p ON u.id = p.wlasciciel WHERE u.email = ?";
$stmt_zajete_miejsce = $conn->prepare($query_zajete_miejsce);
$stmt_zajete_miejsce->bind_param("s", $user_email);
$stmt_zajete_miejsce->execute();
$result_zajete_miejsce = $stmt_zajete_miejsce->get_result();
$stmt_zajete_miejsce->close();

if ($result_zajete_miejsce->num_rows == 1) {
    $row_zajete_miejsce = $result_zajete_miejsce->fetch_assoc();
    $zajete_miejsce_bajty = $row_zajete_miejsce['zajete_miejsce']; // Suma zajętego
miejsca w bajtach

    // Konwersja bajtów na megabajty
    $zajete_miejsce_MB = round($zajete_miejsce_bajty / (1024 * 1024), 2); //
Zaokrąglamy do dwóch miejsc po przecinku
}

// Pobranie miejsca użytkownika
$query = "SELECT miejsce FROM uzytkownicy WHERE email = ?";
$stmt = $conn->prepare($query);
$stmt->bind_param("s", $user_email);
$stmt->execute();
$result = $stmt->get_result();
$miejsce_uzytkownika = $result->fetch_assoc()['miejsce']; // Pobranie miejsca
użytkownika
$stmt->close();

// Sprawdź czy nie został przekroczony limit miejsca
if (round($file_size / (1024 * 1024), 2) + $zajete_miejsce_MB < $miejsce_uzytkownika) {
    $file_content = file_get_contents($file_temp);

    // Pobranie ID użytkownika na podstawie adresu e-mail
    $user_email = $_SESSION['email'];
    $query_user = "SELECT id FROM uzytkownicy WHERE email = ?";
    $stmt_user = $conn->prepare($query_user);
    $stmt_user->bind_param("s", $user_email);
    $stmt_user->execute();
    $result_user = $stmt_user->get_result();

    if ($result_user->num_rows == 1) {
        $row_user = $result_user->fetch_assoc();
        $owner_id = $row_user['id'];

        // Zakodowanie nazwy pliku
        $file_name_encoded = urlencode($file_name);

        // Wstawienie pliku do bazy danych
        $query = "INSERT INTO pliki (nazwa, plik, wlasciciel, bajty) VALUES (?, ?, ?,
?);";

        $stmt = $conn->prepare($query);

```

```

        $stmt->bind_param("ssii", $file_name_encoded, $file_content, $owner_id,
$file_size);
        $stmt->execute();

        // Zamknięcie połączenia z bazą danych
        $stmt->close();
    } else {
        header("location: ../user.php?error=2");
    }

    // Zamknięcie połączenia z bazą danych
    $stmt_user->close();
    $conn->close();

    // Przekierowanie z powrotem na stronę użytkownika
    header("Location: ../user.php");
    exit();
} else {
    header("location: ../user.php?error=4");
}
}
?>

```

Opis:

Walidacja danych: Skrypt zaczyna się od sprawdzenia metody żądania (POST), obecności przesłanego pliku i sesji zalogowanego użytkownika.

Przetwarzanie plików: Skrypt obsługuje przesyłane pliki, pobierając ich nazwę, tymczasową ścieżkę i rozmiar. Sprawdza, czy przesłany plik istnieje, a także czy nie przekracza dozwolonego miejsca użytkownika na serwerze.

Interakcja z bazą danych: Wykorzystuje połączenie z bazą danych do pobrania i aktualizacji informacji o użytkowniku i jego plikach. Obsługuje błędy połączenia i wykonuje zapytania do bazy danych.

Zarządzanie błędami: W przypadku wystąpienia błędów, użytkownik jest przekierowywany z odpowiednim komunikatem błędu.

haslo.php

Obsługuje proces zmiany hasła.

```

<!--
    haslo.php
    skrypt odpowiedzialny za ustawienie hasła
-->
<?php
// Sprawdzenie, czy dane zostały przesłane z formularza
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_GET['kod'])) {
    // Pobranie danych z formularza
    $haslo = $_POST['haslo'];
    $kod = $_GET['kod'];

    // Połączenie z bazą danych
    include('baza.php');
    $polaczenie = mysqli_connect($host, $uzytkownik_bd, $haslo_bd, $bd);
}

```

```

// Sprawdzenie, czy udało się połączyć z bazą danych
if (!$polaczenie) {
    die("Błąd połączenia z bazą danych: " . mysqli_connect_error());
}

// Pobranie emaila użytkownika na podstawie kodu z tabeli rejestracja
$query = "SELECT email FROM rejestracja WHERE kod='$kod'";
$result = mysqli_query($polaczenie, $query);

if ($result && mysqli_num_rows($result) == 1) {
    $row = mysqli_fetch_assoc($result);
    $email = $row['email'];

    // Wstawienie nowego hasła i emaila użytkownika do tabeli uzytkownicy
    // Sprawdzenie, czy adres e-mail istnieje w bazie danych
    $checkQuery = "SELECT * FROM uzytkownicy WHERE email = '$email'";
    $checkResult = mysqli_query($polaczenie, $checkQuery);

    if (mysqli_num_rows($checkResult) > 0) {
        // Adres e-mail istnieje - wykonaj zapytanie aktualizacji hasła
        $updateQuery = "UPDATE uzytkownicy SET haslo = SHA2('$haslo', 256) WHERE email = '$email'";
        $updateResult = mysqli_query($polaczenie, $updateQuery);
    } else {
        // Adres e-mail nie istnieje - wykonaj zapytanie dodania nowego użytkownika
        $insertQuery = "INSERT INTO uzytkownicy (email, haslo) VALUES ('$email', SHA2('$haslo', 256))";
        $insertResult = mysqli_query($polaczenie, $insertQuery);
    }

    if ($insertResult || $updateResult) {
        // Usunięcie rekordu z tabeli rejestracja
        $deleteQuery = "DELETE FROM rejestracja WHERE kod='$kod'";
        $deleteResult = mysqli_query($polaczenie, $deleteQuery);
        if ($deleteResult) {
            header("location: ../index.php?login=1");
        } else {
            echo "Wystąpił błąd podczas usuwania rekordu z tabeli rejestracja.";
        }
    } else {
        echo "Wystąpił błąd podczas zapisywania nowego hasła do bazy danych.";
    }
} else {
    echo "Nie znaleziono użytkownika dla podanego kodu.";
}

// Zamknięcie połączenia z bazą danych
mysqli_close($polaczenie);
} else {
    // Jeśli dane nie zostały przesłane, przekierowanie na stronę główną
    header("location: ../index.php");
    exit;
}
?>

```

Opis:

Proces resetowania hasła: Skrypt ten jest uruchamiany po otrzymaniu formularza z nowym hasłem i kodem resetowania. Sprawdza, czy kod jest prawidłowy i powiązany z adresem e-mail.

Aktualizacja lub wstawianie rekordu: Skrypt aktualizuje istniejące hasło lub tworzy nowego użytkownika w przypadku braku istniejącego rekordu, co jest nietypowe i wymagałoby dodatkowego zabezpieczenia lub potwierdzenia.

Zarządzanie błędami: Skrypt obsługuje błędy związane z nieodnalezieniem użytkownika, problemami z bazą danych oraz zapewnia informację o błędach podczas operacji na bazie.

login.php

Zarządza procesem logowania.

```
<!--
    login.php
    skrypt odpowiedzialny za logowanie na stronie
-->
<?php
session_start();

include('baza.php');

// Połączenie z bazą danych
$polaczenie = mysqli_connect($host, $uzytkownik_bd, $haslo_bd, $bd);

// Sprawdzenie czy połączenie udało się nawiązać
if (!$polaczenie) {
    die("Błąd połączenia z bazą danych: " . mysqli_connect_error());
}

// Pobranie adresu IP użytkownika
$ip = $_SERVER['REMOTE_ADDR'];

// Sprawdzenie, czy adres IP użytkownika jest zablokowany w bazie danych
$query = "SELECT * FROM blokada WHERE ip='$ip'";
$result = mysqli_query($polaczenie, $query);

if (mysqli_num_rows($result) > 0) {
    // Adres IP użytkownika znajduje się w tabeli blokada, przekierowanie na stronę główną
    header("location: ../index.php?error=2");
    exit;
}

// Sprawdzenie czy formularz został wysłany
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Pobranie danych z formularza
    $email = $_POST['email'];
    $haslo = $_POST['haslo'];

    // Zapytanie SQL w celu sprawdzenia czy istnieje użytkownik o podanych danych logowania
    $query = "SELECT * FROM uzytkownicy WHERE email=? AND haslo=SHA2(?,256)";

    // Przygotowanie zapytania SQL
    $stmt = mysqli_prepare($polaczenie, $query);

    // Sprawdzenie czy zapytanie zostało prawidłowo przygotowane
    if ($stmt) {
        // Związanie parametrów
        mysqli_stmt_bind_param($stmt, "ss", $email, $haslo);
```

```

        // Wykonanie zapytania
        mysqli_stmt_execute($stmt);

        // Pobranie wyniku zapytania
        $result = mysqli_stmt_get_result($stmt);

        // Sprawdzenie czy znaleziono użytkownika w bazie danych
        if (mysqli_num_rows($result) == 1) {
            // Pobranie danych użytkownika
            $row = mysqli_fetch_assoc($result);

            // Ustawienie zmiennej sesji na true po poprawnym zalogowaniu
            $_SESSION['zalogowany'] = true;

            // Sprawdzenie typu użytkownika i ustawienie odpowiedniej zmiennej sesji
            $_SESSION['typ_uzytkownika'] = $row['admin'] == 1 ? 'admin' : 'user';

            // przypisanie emaila
            $_SESSION['email'] = $email;

            // Przekierowanie na odpowiednią stronę w zależności od typu użytkownika
            if ($_SESSION['typ_uzytkownika'] == 'admin') {
                header("location: ../admin.php");
            } else {
                header("location: ../user.php");
            }
            exit; // Warto zakończyć dalsze wykonywanie skryptu po przekierowaniu
        } else {
            // Jeżeli nie znaleziono użytkownika, zapisz adres IP użytkownika w tabeli
            $insertQuery = "INSERT INTO blokada (ip) VALUES ('$ip')";
            mysqli_query($polaczenie, $insertQuery);
            // Jeżeli nie znaleziono użytkownika, przekieruj z komunikatem błędu
            header("location: ../index.php?error=1");
            exit; // Warto zakończyć dalsze wykonywanie skryptu po przekierowaniu
        }
    } else {
        // Błąd przygotowania zapytania
        die("Błąd przygotowania zapytania SQL: " . mysqli_error($polaczenie));
    }
}

// Zamknięcie połączenia z bazą danych
mysqli_close($polaczenie);
?>

```

Opis:

Autentykacja: Skrypt obsługuje formularz logowania, pobierając dane i sprawdzając je z bazą danych. Używa bezpiecznej metody `password_verify` do sprawdzenia hasła.

Sesje: Ustawia sesję użytkownika, jeśli dane są prawidłowe, co pozwala na utrzymanie zalogowanego stanu podczas przeglądania.

Zarządzanie błędami: W przypadku błędów autentykacji wyświetla komunikat o błędzie, co informuje użytkownika o nieudanym logowaniu.

plik.php

Obsługuje pobieranie udostępnionych plików

```
<!--
    plik.php
    skrypt odpowiedzialny za pobieranie udostępnionych plików
-->
<?php
require_once('baza.php');

if ($_SERVER["REQUEST_METHOD"] == "GET" && isset($_GET["kod"])) {
    $kod = $_GET["kod"];

    // Połączenie z bazą danych
    $conn = new mysqli($host, $uzytkownik_bd, $haslo_bd, $bd);

    // Sprawdzanie połączenia
    if ($conn->connect_error) {
        die("Błąd połączenia z bazą danych: " . $conn->connect_error);
    }

    // Znalezienie pliku na podstawie kodu
    $query = "SELECT nazwa, plik FROM pliki WHERE kod = ?";
    $stmt = $conn->prepare($query);
    $stmt->bind_param("s", $kod);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows == 1) {
        $row = $result->fetch_assoc();
        $nazwa_pliku = $row['nazwa'];
        $plik = $row['plik'];

        // Ustawienie nagłówków do pobrania pliku
        header("Content-Disposition: attachment; filename=" . $nazwa_pliku);
        header("Content-Type: application/octet-stream");
        header("Content-Length: " . strlen($plik));

        // Wyświetlenie zawartości pliku
        echo $plik;
    } else {
        // Jeśli nie znaleziono pliku dla podanego kodu, zwróć odpowiedni komunikat błędu
        die("Nie można znaleźć pliku dla podanego kodu.");
    }

    // Zamknięcie połączenia z bazą danych
    $stmt->close();
    $conn->close();

    exit();
}
?>
```

Opis:

Dostęp do pliku: Skrypt pozwala na pobranie pliku na podstawie unikalnego kodu, który jest przypisany do pliku w bazie danych. Jest to często stosowana metoda do udostępniania plików bez potrzeby logowania.

Bezpieczeństwo: Użycie przygotowanych zapytań zabezpiecza przed SQL Injection, a sprawdzanie kodu dostępu zapewnia, że tylko osoby z odpowiednim kodem mogą pobrać plik.

Obsługa błędów: W przypadku, gdy plik nie zostanie znaleziony, skrypt zwraca błąd i kończy działanie.

pobierz_plik.php

Umożliwia pobieranie plików przez użytkownika.

```
<!--
    pobierz_plik.php
    skrypt odpowiedzialny za pobieranie plików
-->
<?php
session_start();

// Sprawdzenie czy użytkownik jest zalogowany
if (!isset($_SESSION['zalogowany']) || $_SESSION['zalogowany'] !== true) {
    // Jeżeli nie, przekieruj na stronę logowania
    header("location: ../index.php");
    exit;
}

// Sprawdzenie czy użytkownik ma uprawnienia użytkownika
if (!isset($_SESSION['typ_uzytkownika']) || $_SESSION['typ_uzytkownika'] !== 'user') {
    // Jeżeli nie, przekieruj na stronę główną
    header("location: ../index.php");
    exit;
}

// Sprawdzenie czy żądanie jest typu POST i czy zawiera ID pliku
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['plik_id'])) {
    // Pobranie ID pliku
    $file_id = $_POST['plik_id'];

    // Połączenie z bazą danych
    require_once('baza.php');
    $conn = new mysqli($host, $uzytkownik_bd, $haslo_bd, $bd);

    // Sprawdzenie połączenia
    if ($conn->connect_error) {
        die("Błąd połączenia z bazą danych: " . $conn->connect_error);
    }
    $user_email = $_SESSION['email'];
    // Pobranie ID użytkownika na podstawie adresu e-mail
    $query_user = "SELECT id FROM uzytkownicy WHERE email = ?";
    $stmt_user = $conn->prepare($query_user);
    $stmt_user->bind_param("s", $user_email);
    $stmt_user->execute();
    $result_user = $stmt_user->get_result();

    if ($result_user->num_rows == 1) {
        $row_user = $result_user->fetch_assoc();
    }
}
```

```

$user_id = $row_user['id'];

// Pobranie informacji o pliku
$query_file = "SELECT nazwa, plik FROM pliki WHERE id = ? AND wlasciciel = ?";
$stmt_file = $conn->prepare($query_file);
$stmt_file->bind_param("ii", $file_id, $user_id);
$stmt_file->execute();
$result_file = $stmt_file->get_result();

if ($result_file->num_rows == 1) {
    $row_file = $result_file->fetch_assoc();
    $file_name = $row_file['nazwa'];
    $file_content = $row_file['plik'];

    // Ustawienie nagłówków dla pobierania pliku
    $file_name = urldecode($file_name);
    header("Content-Disposition: attachment; filename=\"$file_name\"");
    header("Content-Type: application/octet-stream");
    header("Content-Length: " . strlen($file_content));

    // Wyślij zawartość pliku do użytkownika
    echo $file_content;
    exit();
} else {
    header("location: ../user.php?error=1");
}

// Zamknięcie połączenia z bazą danych
$stmt_file->close();
} else {
    header("location: ../user.php?error=2");
}

// Zamknięcie połączenia z bazą danych
$stmt_user->close();
$conn->close();
} else {
    header("location: ../user.php?error=3");
}
?>

```

Opis:

Weryfikacja użytkownika: Skrypt sprawdza, czy użytkownik jest zalogowany i czy ma odpowiednie uprawnienia do pobierania plików.

Pobieranie plików: Skrypt pobiera plik przypisany do użytkownika z bazy danych i umożliwia jego pobranie, ustawiając odpowiednie nagłówki HTTP.

przestan_udostepniac.php

Pozwala na zakończenie udostępniania pliku.

```

<!--
    przestan_udostepniac.php
    skrypt odpowiedzialny za zaprzestanie udostępniania pliku
-->
<?php

```



```

session_start();
require_once('baza.php');

if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST["przestan_udostepniac"]) &&
isset($_POST["plik_id"]) && isset($_SESSION['email'])) {
    $plik_id = $_POST["plik_id"];

    // Połączenie z bazą danych
    $conn = new mysqli($host, $uzytkownik_bd, $haslo_bd, $bd);

    // Sprawdzanie połączenia
    if ($conn->connect_error) {
        die("Błąd połączenia z bazą danych: " . $conn->connect_error);
    }

    // Pobranie ID użytkownika na podstawie adresu e-mail
    $user_email = $_SESSION['email'];
    $query_user = "SELECT id FROM uzytkownicy WHERE email = ?";
    $stmt_user = $conn->prepare($query_user);
    $stmt_user->bind_param("s", $user_email);
    $stmt_user->execute();
    $result_user = $stmt_user->get_result();

    if ($result_user->num_rows == 1) {
        $row_user = $result_user->fetch_assoc();
        $user_id = $row_user['id'];

        // Usunięcie kodu udostępniania z pliku
        $query_update = "UPDATE pliki SET kod = NULL WHERE id = ? AND wlasciciel = ?";
        $stmt_update = $conn->prepare($query_update);
        $stmt_update->bind_param("ii", $plik_id, $user_id);
        $stmt_update->execute();

        // Zamknięcie połączenia z bazą danych
        $stmt_update->close();
    } else {
        header("location: ../user.php?error=2");
    }

    // Zamknięcie połączenia z bazą danych
    $stmt_user->close();
    $conn->close();

    // Przekierowanie na stronę użytkownika
    header("Location: ../user.php");
    exit();
}
?>

```

Opis:

Autoryzacja i sesja: Podobnie jak inne skrypty, sprawdza czy użytkownik jest zalogowany i czy posiada odpowiednie uprawnienia.

Zarządzanie dostępem do plików: Skrypt umożliwia anulowanie udostępnienia pliku przez usunięcie kodu, który umożliwia dostęp. Wykorzystuje bezpieczne metody przygotowania zapytań do bazy danych, by uniknąć iniekcji SQL.

przypomnij.php

Wysyła e-mail do resetowania hasła.

```
<!--
    przypomnij.php
    skrypt odpowiedzialny za wysłanie emaila do użytkownika jeżeli ten zapomniał hasła
-->
<?php
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

// Include Composer's autoloader
require 'vendor/autoload.php';

// Połączenie z bazą danych
include('baza.php');

// Połączenie z bazą danych
$polaczenie = mysqli_connect($host, $uzytkownik_bd, $haslo_bd, $bd);

// Sprawdzenie czy połączenie udało się nawiązać
if (!$polaczenie) {
    die("Błąd połączenia z bazą danych: " . mysqli_connect_error());
}

// Pobranie adresu IP użytkownika
$ip = $_SERVER['REMOTE_ADDR'];

// Sprawdzenie, czy adres IP użytkownika jest zablokowany w bazie danych
$query = "SELECT * FROM blokada2 WHERE ip='$ip'";
$result = mysqli_query($polaczenie, $query);

if (mysqli_num_rows($result) > 0) {
    // Adres IP użytkownika znajduje się w tabeli blokada, przekierowanie na stronę główną
    header("location: ../przypomnij.php?error=2");
    exit;
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Pobranie adresu e-mail z formularza
    $email = $_POST['email'];

    // Sprawdzenie, czy podany e-mail istnieje już w bazie danych
    $query = "SELECT * FROM uzytkownicy WHERE email = ?";
    if ($stmt = mysqli_prepare($polaczenie, $query)) {
        mysqli_stmt_bind_param($stmt, "s", $email);
        mysqli_stmt_execute($stmt);
        mysqli_stmt_store_result($stmt);
        $emailExists = mysqli_stmt_num_rows($stmt) > 0;
        mysqli_stmt_close($stmt);
    } else {
        echo "Błąd przygotowania zapytania SQL: " . mysqli_error($polaczenie);
        exit;
    }
    mysqli_close($polaczenie);
    if (!$emailExists) {
        header("Location: ../przypomnij.php?error=1");
    }
}
```

```

        exit;
    }

    // Tworzenie nowej instancji PHPMailer
    $mail = new PHPMailer(true); // true oznacza, że zostanie wyrzucony wyjątek w przypadku
    błędu

    try {
        // Konfiguracja serwera SMTP
        $mail->isSMTP();
        $mail->Host = 'smtp-relay.brevo.com';
        $mail->SMTPAuth = true;
        $mail->Username = 'j.martynski2002@gmail.com'; // Twój adres e-mail
        $mail->Password = 'XgH207kqGEKS8Cah'; // Twoje hasło do konta
        $mail->SMTPSecure = 'auto';
        $mail->Port = 587;

        // Konfiguracja nadawcy i odbiorcy
        $mail->setFrom('j.martynski2002@gmail.com', 'noreply@liqteam.pl'); // Twój adres e-
        mail jako nadawca
        $mail->addAddress($email); // Adres e-mail odbiorcy

        // Treść wiadomości
        $mail->isHTML(true); // Ustawienie formatu wiadomości na HTML

        // Wygenerowanie losowego kodu
        $kod = bin2hex(random_bytes(10)); // Długość 20 znaków w formacie szesnastkowym

        // Dodanie danych do tabeli rejestracja
        $polaczenie = mysqli_connect($host, $uzytkownik_bd, $haslo_bd, $bd);
        $query = "INSERT INTO rejestracja (email, kod) VALUES (?, ?)";

        // Przygotowanie i wykonanie instrukcji przygotowanej
        if ($stmt = mysqli_prepare($polaczenie, $query)) {
            // Związanie parametrów
            mysqli_stmt_bind_param($stmt, "ss", $email, $kod);

            // Wykonanie zapytania
            mysqli_stmt_execute($stmt);

            // Zwolnienie wyniku zapytania
            mysqli_stmt_close($stmt);
        } else {
            // Obsługa błędu przygotowania zapytania
            echo "Błąd przygotowania zapytania SQL: " . mysqli_error($polaczenie);
        }

        // Zamknięcie połączenia z bazą danych (przeniesione na koniec skryptu)
        // mysqli_close($polaczenie);

        // Treść wiadomości
        $mail->CharSet = 'UTF-8';
        $mail->Encoding = 'base64';
        $mail->Subject = 'odzyskiwanie hasła';
        $mail->Body = "Zmien swoje hasło tutaj:<br>
        <a
href='$moja_strona'.\"haslo.php?kod=$kod\">$moja_strona\".\"haslo.php?kod=$kod</a><br>
        Link wazny przez godzine.";
    }

```

```

        // Wysłanie wiadomości
        $mail->send();
        echo 'Wiadomość e-mail została wysłana na adres: ' . $email;
        $insertQuery = "INSERT INTO blokada2 (ip) VALUES ('$ip')";
        mysqli_query($polaczenie, $insertQuery);

    } catch (Exception $e) {
        echo 'Wystąpił błąd podczas wysyłania wiadomości: ', $mail->ErrorInfo;
    }

    // Zamknięcie połączenia z bazą danych (przeniesione na koniec skryptu)
    mysqli_close($polaczenie);
}
?>

```

Opis:

Reset hasła: Skrypt pozwala na resetowanie hasła, sprawdzając, czy podany e-mail istnieje w bazie danych, a następnie generuje unikalny token resetujący.

Bezpieczeństwo: Używa bezpiecznych połączeń z bazą danych i metody przygotowanych zapytań do zapobiegania atakom SQL Injection.

Komunikacja e-mail: Wysyła e-mail z linkiem resetującym, umożliwiając użytkownikowi zresetowanie hasła.

rejestracja.php

Obsługuje proces rejestracji nowych użytkowników.

```

<!--
    rejestracja.php
    skrypt odpowiedzialny za wysłanie emaila do użytkownika jeżeli ten chce się
zarejestrować
-->
<?php
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

// Include Composer's autoloader
require 'vendor/autoload.php';

// Połączenie z bazą danych
include('baza.php');

// Połączenie z bazą danych
$polaczenie = mysqli_connect($host, $uzytkownik_bd, $haslo_bd, $bd);

// Sprawdzenie czy połączenie udało się nawiązać
if (!$polaczenie) {
    die("Błąd połączenia z bazą danych: " . mysqli_connect_error());
}

// Pobranie adresu IP użytkownika
$ip = $_SERVER['REMOTE_ADDR'];

// Sprawdzenie, czy adres IP użytkownika jest zablokowany w bazie danych
$query = "SELECT * FROM blokada2 WHERE ip='$ip'";

```

```

$result = mysqli_query($polaczenie, $query);

if (mysqli_num_rows($result) > 0) {
    // Adres IP użytkownika znajduje się w tabeli blokada, przekierowanie na stronę główną
    header("location: ../rejestracja.php?error=2");
    exit;
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Pobranie adresu e-mail z formularza
    $email = $_POST['email'];

    // Sprawdzenie, czy podany e-mail istnieje już w bazie danych
    $query = "SELECT * FROM uzytkownicy WHERE email = ?";
    if ($stmt = mysqli_prepare($polaczenie, $query)) {
        mysqli_stmt_bind_param($stmt, "s", $email);
        mysqli_stmt_execute($stmt);
        mysqli_stmt_store_result($stmt);
        $emailExists = mysqli_stmt_num_rows($stmt) > 0;
        mysqli_stmt_close($stmt);
    } else {
        echo "Błąd przygotowania zapytania SQL: " . mysqli_error($polaczenie);
        exit;
    }
    if ($emailExists) {
        header("Location: ../rejestracja.php?error=1");
        exit;
    }

    // Tworzenie nowej instancji PHPMailer
    $mail = new PHPMailer(true); // true oznacza, że zostanie wyrzucony wyjątek w przypadku
    błędu

    try {
        // Konfiguracja serwera SMTP
        $mail->isSMTP();
        $mail->Host = 'smtp-relay.brevo.com';
        $mail->SMTPAuth = true;
        $mail->Username = 'j.martynski2002@gmail.com'; // Twój adres e-mail
        $mail->Password = 'XgH207kqGEKS8Cah'; // Twoje hasło do konta
        $mail->SMTPSecure = 'auto';
        $mail->Port = 587;

        // Konfiguracja nadawcy i odbiorcy
        $mail->setFrom('j.martynski2002@gmail.com', 'noreply@liqteam.pl'); // Twój adres e-
        mail jako nadawca
        $mail->addAddress($email); // Adres e-mail odbiorcy

        // Treść wiadomości
        $mail->isHTML(true); // Ustawienie formatu wiadomości na HTML

        // Wygenerowanie losowego kodu
        $kod = bin2hex(random_bytes(10)); // Długość 20 znaków w formacie szesnastkowym

        // Dodanie danych do tabeli rejestracja
        $query = "INSERT INTO rejestracja (email, kod) VALUES (?, ?)";

        // Przygotowanie i wykonanie instrukcji przygotowanej

```

```

if ($stmt = mysqli_prepare($polaczenie, $query)) {
    // Związanie parametrów
    mysqli_stmt_bind_param($stmt, "ss", $email, $kod);

    // Wykonanie zapytania
    mysqli_stmt_execute($stmt);

    // Zwolnienie wyniku zapytania
    mysqli_stmt_close($stmt);
} else {
    // Obsługa błędu przygotowania zapytania
    echo "Błąd przygotowania zapytania SQL: " . mysqli_error($polaczenie);
}

// Treść wiadomości
$mail->CharSet = 'UTF-8';
$mail->Encoding = 'base64';
$mail->Subject = 'rejestracja';
$mail->Body = "Dziękujemy za rejestrację na naszej stronie!<br>
Zeby potwierdzić swoje konto kliknij ten link:<br>
<a
href='$moja_strona'.\"haslo.php?kod=$kod\">$moja_strona\".\"haslo.php?kod=$kod</a><br>
Link ważny przez godzinę.";

// Wysłanie wiadomości
$mail->send();
echo 'Wiadomość e-mail została wysłana na adres: ' . $email;
$insertQuery = "INSERT INTO blokada2 (ip) VALUES ('$ip')";
mysqli_query($polaczenie, $insertQuery);

} catch (Exception $e) {
    echo 'Wystąpił błąd podczas wysyłania wiadomości: ', $mail->ErrorInfo;
}
}
?>

```

Opis:

Rejestracja użytkownika: Skrypt ten sprawdza, czy podany adres e-mail już istnieje w systemie. Jeśli nie, to wysyła e-mail z linkiem do aktywacji, który zawiera wygenerowany kod.

Zarządzanie wyjątkami: PHPMailer jest używany do obsługi wysyłki e-maili, co pozwala na bezpieczne i elastyczne zarządzanie procesem wysyłania wiadomości e-mail.

Bezpieczeństwo: Użycie przygotowanych zapytań do bazy danych minimalizuje ryzyko iniekcji SQL.

udostepnij_plik.php

Zarządza udostępnianiem plików przez użytkowników.

```

<!--
    udostepnij_plik.php
    skrypt odpowiedzialny za udostepnianie plików
-->
<?php
session_start();
require_once('baza.php');

```

```

if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST["udostepnij"]) &&
isset($_POST["plik_id"]) && isset($_SESSION['email'])) {
    $plik_id = $_POST["plik_id"];

    // Połączenie z bazą danych
    $conn = new mysqli($host, $uzytkownik_bd, $haslo_bd, $bd);

    // Sprawdzanie połączenia
    if ($conn->connect_error) {
        die("Błąd połączenia z bazą danych: " . $conn->connect_error);
    }

    // Pobranie ID użytkownika na podstawie adresu e-mail
    $user_email = $_SESSION['email'];
    $query_user = "SELECT id FROM uzytkownicy WHERE email = ?";
    $stmt_user = $conn->prepare($query_user);
    $stmt_user->bind_param("s", $user_email);
    $stmt_user->execute();
    $result_user = $stmt_user->get_result();

    if ($result_user->num_rows == 1) {
        $row_user = $result_user->fetch_assoc();
        $user_id = $row_user['id'];

        // Generowanie kodu do udostępnienia
        $kod = generateRandomString(20);

        // Aktualizacja kodu w bazie danych
        $query_update = "UPDATE pliki SET kod = ? WHERE id = ? AND wlasciciel = ?";
        $stmt_update = $conn->prepare($query_update);
        $stmt_update->bind_param("sii", $kod, $plik_id, $user_id);
        $stmt_update->execute();

        // Zamknięcie połączenia z bazą danych
        $stmt_update->close();
    } else {
        header("location: ../user.php?error=2");
    }

    // Zamknięcie połączenia z bazą danych
    $stmt_user->close();
    $conn->close();

    // Przekierowanie na stronę użytkownika
    header("Location: ../user.php");
    exit();
}

// Funkcja do generowania losowego ciągu znaków
function generateRandomString($length = 10) {
    $characters = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
    $charactersLength = strlen($characters);
    $randomString = '';
    for ($i = 0; $i < $length; $i++) {
        $randomString .= $characters[rand(0, $charactersLength - 1)];
    }
    return $randomString;
}

```

```
?>
```

Opis:

Udostępnianie plików: Skrypt pozwala na udostępnianie plików poprzez generowanie unikalnego kodu, który jest przypisywany do pliku w bazie danych.

Zarządzanie uprawnieniami i sesjami: Podobnie jak inne skrypty, wymaga zalogowania i weryfikuje uprawnienia użytkownika.

Generowanie kodu: Wykorzystuje funkcję `generateRandomString`, aby stworzyć unikalny kod, który następnie jest przypisywany do pliku.

usun_plik.php

Pozwala na usunięcie pliku z systemu.

```
<!--
    usun_plik.php
    skrypt odpowiedzialny za usuwanie plików
-->
<?php
session_start();

// Sprawdzenie czy użytkownik jest zalogowany
if (!isset($_SESSION['zalogowany']) || $_SESSION['zalogowany'] !== true) {
    // Jeżeli nie, przekieruj na stronę logowania
    header("location: ../index.php");
    exit;
}

// Sprawdzenie czy użytkownik ma uprawnienia użytkownika
if (!isset($_SESSION['typ_uzytkownika']) || $_SESSION['typ_uzytkownika'] !== 'user') {
    // Jeżeli nie, przekieruj na stronę główną
    header("location: ../index.php");
    exit;
}

// Sprawdzenie czy żądanie jest typu POST i czy zawiera ID pliku do usunięcia
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['plik_id'])) {
    // Pobranie ID pliku do usunięcia
    $plik_id = $_POST['plik_id'];

    // Połączenie z bazą danych
    require_once('baza.php');
    $conn = new mysqli($host, $uzytkownik_bd, $haslo_bd, $bd);

    // Sprawdzenie połączenia
    if ($conn->connect_error) {
        die("Błąd połączenia z bazą danych: " . $conn->connect_error);
    }

    // Usunięcie pliku z bazy danych
    $query_delete = "DELETE FROM pliki WHERE id = ?";
    $stmt_delete = $conn->prepare($query_delete);
    $stmt_delete->bind_param("i", $plik_id);
    $stmt_delete->execute();
}
```



```

    // Zamknięcie połączenia z bazą danych
    $stmt_delete->close();
    $conn->close();

    // Przekierowanie na stronę użytkownika
    header("Location: ../user.php");
    exit();
} else {
    // Jeżeli nie, przekieruj na stronę użytkownika
    header("Location: ../user.php");
    exit();
}
?>

```

Opis:

Walidacja sesji: Skrypt zaczyna się od sprawdzenia, czy użytkownik jest zalogowany i czy ma odpowiednie uprawnienia do wykonania operacji usuwania pliku.

Bezpieczeństwo: Skrypt korzysta z metody POST, co minimalizuje ryzyko nieautoryzowanego usunięcia przez manipulację URL.

Interakcja z bazą danych: Skrypt tworzy połączenie z bazą danych, wykonuje zapytanie SQL na usunięcie wybranego pliku po jego ID, a następnie zamyka połączenie. W przypadku błędów połączenia wyświetlany jest komunikat o błędzie.

Przekierowania: Po pomyślnym usunięciu pliku użytkownik jest przekierowywany z powrotem na stronę użytkownika.

usun_uzytkownika.php

Umożliwia usunięcie konta użytkownika przez administratora.

```

<!--
    usun_uzytkownika.php
    skrypt odpowiedzialny za usuwanie użytkowników
-->
<?php
// Połączenie z bazą danych
require_once('baza.php');
$conn = new mysqli($host, $uzytkownik_bd, $haslo_bd, $bd);

// Sprawdzenie połączenia
if ($conn->connect_error) {
    die("Błąd połączenia z bazą danych: " . $conn->connect_error);
}

if(isset($_POST['usun'])) {
    // Pobranie emaila użytkownika do usunięcia
    $email = $_POST['email'];

    // Usunięcie wszystkich plików użytkownika
    $query_usun_pliki = "DELETE FROM pliki WHERE wlasciciel IN (SELECT id FROM uzytkownicy
WHERE email = ?)";
    $stmt_usun_pliki = $conn->prepare($query_usun_pliki);
    $stmt_usun_pliki->bind_param("s", $email);
    $stmt_usun_pliki->execute();
}

```

```

$stmt_usun_pliki->close();

// Usunięcie użytkownika
$query_usun_uzytkownika = "DELETE FROM uzytkownicy WHERE email = ?";
$stmt_usun_uzytkownika = $conn->prepare($query_usun_uzytkownika);
$stmt_usun_uzytkownika->bind_param("s", $email);
$stmt_usun_uzytkownika->execute();

// Informacja o powodzeniu lub błędzie
if ($stmt_usun_uzytkownika->affected_rows > 0) {
    header("location: ../admin.php");
} else {
    header("location: ../admin.php");
}

// Zamknięcie połączenia
$stmt_usun_uzytkownika->close();
$conn->close();
}
?>

```

Opis:

Usuwanie danych użytkownika: Ten skrypt bezpiecznie usuwa wszystkie pliki należące do użytkownika, a następnie usuwa samego użytkownika z bazy danych.

Zarządzanie zasobami: Po zakończeniu operacji na bazie danych, połączenia są zamykane, co jest dobrym praktyką zarządzania zasobami.

Przekierowania: Po wykonaniu operacji, administrator jest przekierowywany z powrotem na stronę zarządzania, gdzie może zobaczyć wynik operacji.

zmien_miejsce.php

Zmienia limit przestrzeni dyskowej przypisany do użytkownika.

```

<!--
    zmien_miejsce.php
    skrypt odpowiedzialny za zmianę miejsca które przysługuje użytkownikowi
-->
<?php
// Połączenie z bazą danych
require_once('baza.php');
$conn = new mysqli($host, $uzytkownik_bd, $haslo_bd, $bd);

// Sprawdzenie połączenia
if ($conn->connect_error) {
    die("Błąd połączenia z bazą danych: " . $conn->connect_error);
}

if(isset($_POST['zmien'])) {
    // Pobranie nowej wartości z formularza
    $nowe_miejsce = $_POST['nowe_miejsce'];
    $email = $_POST['email']; // Dodane pole ukryte z wartością email

    // Aktualizacja wartości w bazie danych
    $query_update = "UPDATE uzytkownicy SET miejsce = ? WHERE email = ?";
    $stmt_update = $conn->prepare($query_update);
}

```

```

$stmt_update->bind_param("ss", $nowe_miejsce, $email);
$stmt_update->execute();

// Informacja o powodzeniu lub błędzie
if ($stmt_update->affected_rows > 0) {
    header("location: ../admin.php");
} else {
    header("location: ../admin.php");
}

// Zamknięcie połączenia
$stmt_update->close();
$conn->close();
}
?>

```

Opis:

Aktualizacja limitów: Skrypt pozwala na aktualizację limitów przestrzeni dyskowej przydzielonej użytkownikom, co jest istotne dla zarządzania zasobami w środowisku chmurowym.

Bezpieczne aktualizacje: Używa bezpiecznych zapytań przygotowanych do aktualizacji danych, co minimalizuje ryzyko błędów i ataków na bazę danych.

Feedback: Administrator jest informowany o wyniku operacji przez przekierowanie na odpowiednią stronę zarządzania.

5.Instalacja

Użytkownik:

Znajdując się na stronie głównej należy założyć konto. Robi się to poprzez wejście w link zarejestruj się gdzie następnie podaje się swojego emaila na którego przyjdzie link aktywacyjny. Po kliknięciu w link nastąpi przekierowanie na stronę gdzie należy podać nowe hasło do konta. Po tym kroku wystarczy się zalogować i konto jest gotowe.

Firma:

Należy pobrać kod źródłowy (2 foldery: projekt i mysql).

Zawartość mysql należy zaimportować do bazy danych MySQL (baza.sql, uzytkownik.sql i events.sql)

Folder projekt należy wrzucić do folderu ze skonfigurowaną stroną Apache.

Zależnie od nazwy domeny należy skonfigurować ją w skryptach baza.php i baza.php wersja ukr.

6.Testy

W celu przetestowania funkcjonalności projektu należy uruchomić projekt Testy napisany w Python.

Plik testy.py znajdujący się w folderze src testuje poszczególne części strony.

W każdym pliku znajduje się wyjaśnienie do czego test jest przeznaczony

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import sys, os, time
from urllib.parse import urlparse, parse_qs
```

wszystkie potrzebne importy żeby skrypt zadziałał

```
service = Service(executable_path="chromedriver.exe")
driver = webdriver.Chrome(service=service)

driver.get("https://www.jakubmartynski.pl/projekt/index.php")
```

start przeglądarki i strony

```
try:
```

wszystko wsadzone w try ponieważ jeżeli jakiś test nie zadziała to nie pojawiają się terrory

```
# Zaloguj się używając danych j.martynski2002@gmail.com #admin
driver.find_element(By.NAME, "email").send_keys("j.martynski2002@gmail.com")
driver.find_element(By.NAME, "haslo").send_keys("NaDzIeJa_76!")
driver.find_element(By.CLASS_NAME, "btn-login").click()
# Poczekaj na przejście do strony admina po zalogowaniu
WebDriverWait(driver, 10).until(EC.title_contains("Strona administratora"))
print("Udało się przejść na stronę administratora (admin.php)")
# Wyloguj się
driver.find_element(By.NAME, "wyloguj").click()
# Poczekaj na przejście do strony logowania po wylogowaniu
WebDriverWait(driver, 10).until(EC.title_contains("Logowanie"))
print("Pomyślnie wylogowano")

# Zaloguj się używając danych 109266@g.elearn.uz.zgora.pl #user
driver.find_element(By.NAME, "email").send_keys("109266@g.elearn.uz.zgora.pl")
driver.find_element(By.NAME, "haslo").send_keys("NaDzIeJa_76!")
driver.find_element(By.CLASS_NAME, "btn-login").click()
# Poczekaj na przejście do strony użytkownika po zalogowaniu
WebDriverWait(driver, 10).until(EC.title_contains("Strona użytkownika"))
print("Udało się przejść na stronę użytkownika (user.php)")
# Wyloguj się
driver.find_element(By.NAME, "wyloguj").click()
# Poczekaj na przejście do strony logowania po wylogowaniu
WebDriverWait(driver, 10).until(EC.title_contains("Logowanie"))
print("Pomyślnie wylogowano")

# Sprawdź, czy na stronie znajduje się link do rejestracji
link_rejestracja = driver.find_element(By.XPATH, "//a[@href='rejestracja.php']")
link_rejestracja.click()
WebDriverWait(driver, 10).until(EC.title_contains("Rejestracja"))
print("Udało się przejść na stronę rejestracji (rejestracja.php)")

# Powrót do strony głównej
driver.back()

# Sprawdź, czy na stronie znajduje się link do przypomnienia hasła
link_przypomnij_haslo = driver.find_element(By.XPATH, "//a[@href='przypomnij.php']")
link_przypomnij_haslo.click()
WebDriverWait(driver, 10).until(EC.title_contains("Przypomnienie"))
```

```
print("Udało się przejść na stronę przypomnienia hasła (przypomnij.php)")

# Powrót do strony głównej
driver.back()

# to samo ale po stronie ukraińskiej

# zmień język
driver.find_element(By.NAME, "jezyk").click()
WebDriverWait(driver, 10).until(EC.title_contains("Логін"))
print("Udało się przejść na stronę po ukraińsku (ukr/index.php)")

# Zaloguj się używając danych j.martynski2002@gmail.com #admin
driver.find_element(By.NAME, "email").send_keys("j.martynski2002@gmail.com")
driver.find_element(By.NAME, "haslo").send_keys("NaDzIeJa_76!")
driver.find_element(By.CLASS_NAME, "btn-login").click()
# Poczekaj na przejście do strony admina po zalogowaniu
WebDriverWait(driver, 10).until(EC.title_contains("Сторінка адміністратора"))
print("Udało się przejść na stronę administratora (ukr/admin.php)")
# Wyloguj się
driver.find_element(By.NAME, "wyloguj").click()
# Poczekaj na przejście do strony logowania po wylogowaniu
WebDriverWait(driver, 10).until(EC.title_contains("Логін"))
print("Pomyślnie wylogowano (ukr)")

# Zaloguj się używając danych 109266@g.elearn.uz.zgora.pl #user
driver.find_element(By.NAME, "email").send_keys("109266@g.elearn.uz.zgora.pl")
driver.find_element(By.NAME, "haslo").send_keys("NaDzIeJa_76!")
driver.find_element(By.CLASS_NAME, "btn-login").click()
# Poczekaj na przejście do strony użytkownika po zalogowaniu
WebDriverWait(driver, 10).until(EC.title_contains("Сторінка користувача"))
print("Udało się przejść na stronę użytkownika (ukr/user.php)")
# Wyloguj się
driver.find_element(By.NAME, "wyloguj").click()
# Poczekaj na przejście do strony logowania po wylogowaniu
WebDriverWait(driver, 10).until(EC.title_contains("Логін"))
print("Pomyślnie wylogowano (ukr)")

# Sprawdź, czy na stronie znajduje się link do rejestracji
link_rejestracja = driver.find_element(By.XPATH, "//a[@href='rejestracja.php']")
link_rejestracja.click()
WebDriverWait(driver, 10).until(EC.title_contains("Реєстрація"))
print("Udało się przejść na stronę rejestracji (ukr/rejestracja.php)")

# Powrót do strony głównej
driver.back()

# Sprawdź, czy na stronie znajduje się link do przypomnienia hasła
link_przypomnij_haslo = driver.find_element(By.XPATH, "//a[@href='przypomnij.php']")
link_przypomnij_haslo.click()
WebDriverWait(driver, 10).until(EC.title_contains("Нагадування"))
print("Udało się przejść na stronę przypomnienia hasła (ukr/przypomnij.php)")

# Powrót do strony głównej
driver.back()

driver.find_element(By.NAME, "jezyk").click()
WebDriverWait(driver, 10).until(EC.title_contains("Logowanie"))
```

```

print("Udało się powrócić na stronę logowania po polsku (index.php)")

# Zaloguj się używając danych 109266@g.elearn.uz.zgora.pl #user
driver.find_element(By.NAME, "email").send_keys("109266@g.elearn.uz.zgora.pl")
driver.find_element(By.NAME, "haslo").send_keys("NaDzIeJa_76!")
driver.find_element(By.CLASS_NAME, "btn-login").click()
# Poczekaj na przejście do strony użytkownika po zalogowaniu
WebDriverWait(driver, 10).until(EC.title_contains("Strona użytkownika"))
print("Udało się przejść na stronę użytkownika (user.php)")

# Przesłanie pliku
file_name = "test.txt"
file_path = os.path.abspath(file_name)
with open(file_path, "w") as file:
    file.write("To jest testowy plik.")
driver.find_element(By.ID, "fileInput").send_keys(file_path)
driver.find_element(By.NAME, "submit").click()
# Czekanie, aż nazwa pliku pojawi się na stronie
WebDriverWait(driver, 10).until(lambda driver: file_name in driver.page_source)
print("Plik test.txt został pomyślnie przesłany.")

# Przesłanie pliku
file_name = "test2.txt"
file_path = os.path.abspath(file_name)
with open(file_path, "w") as file:
    file.write("To jest testowy plik.")
driver.find_element(By.ID, "fileInput").send_keys(file_path)
driver.find_element(By.NAME, "submit").click()
# Czekanie, aż nazwa pliku pojawi się na stronie
WebDriverWait(driver, 10).until(lambda driver: file_name in driver.page_source)
print("Plik test2.txt został pomyślnie przesłany.")

# Wyszukiwanie pliku 'test.txt'
search_box = driver.find_element(By.NAME, "fraz")
search_box.send_keys("test.txt")
search_box.submit() # Submit the form
# Czekanie, aż URL zawiera frazę 'test.txt'
WebDriverWait(driver, 10).until(
    lambda d: 'fraz' in parse_qs(urlparse(d.current_url).query) and
    parse_qs(urlparse(d.current_url).query)['fraz'][0] == 'test.txt'
)
# Sprawdzanie, czy 'test2.txt' nie jest wyświetlany
page_source = driver.page_source
if "test2.txt" not in page_source:
    print("Plik 'test2.txt' nie jest widoczny po wyszukaniu 'test.txt', co jest
    zachowaniem oczekiwanym.")
else:
    print("Plik 'test2.txt' jest nadal widoczny, co jest błędem.")

WebDriverWait(driver, 10).until(EC.visibility_of_element_located((By.XPATH,
"//button[text()='Pobierz']")))
# Kliknięcie przycisku Pobierz dla 'test.txt'
download_buttons = driver.find_elements(By.XPATH, "//button[text()='Pobierz']")
if download_buttons:
    download_buttons[0].click()
    print("Przycisk Pobierz został kliknięty dla pliku test.txt")
else:
    print("Nie znaleziono przycisku Pobierz dla pliku test.txt")

```

```

# Ścieżka do folderu Pobrane (zakładając, że jest to standardowa lokalizacja dla
systemu Windows)
download_folder = os.path.join(os.path.expanduser("~"), "Downloads")
# Nazwa pliku do sprawdzenia
file_name = "test.txt"
file_path = os.path.join(download_folder, file_name)
# Oczekaj, aby dać przeglądarce czas na pobranie pliku
time.sleep(2) # Czas w sekundach; dostosuj zależnie od prędkości połączenia
internetowego

# Sprawdź, czy plik istnieje w folderze Pobrane
if os.path.exists(file_path):
    print(f"Plik {file_name} został pobrany.")
    # Usunięcie pliku
    os.remove(file_path)
    print(f"Plik {file_name} został usunięty.")
else:
    print(f"Plik {file_name} nie został znaleziony w folderze Pobrane.")

# # sprawdzenie udostępniania # #

# Znajdowanie przycisku "Usuń" dla przesłanego pliku
delete_button = driver.find_element(By.XPATH, f"//h3[contains(text(),
'{file_name}')] / ancestor::li / following-
sibling::li / form[@action='skrypty/usun_plik.php'] / button")
delete_button.click()
# Sprawdzenie, czy plik został usunięty
WebDriverWait(driver, 10).until(lambda driver: file_name not in driver.page_source)
print("Plik test.txt został pomyślnie usunięty.")
file_name = "test2.txt"
# Znajdowanie przycisku "Usuń" dla przesłanego pliku
delete_button = driver.find_element(By.XPATH, f"//h3[contains(text(),
'{file_name}')] / ancestor::li / following-
sibling::li / form[@action='skrypty/usun_plik.php'] / button")
delete_button.click()
# Sprawdzenie, czy plik został usunięty
WebDriverWait(driver, 10).until(lambda driver: file_name not in driver.page_source)
print("Plik test2.txt został pomyślnie usunięty.")

# Wyloguj się
driver.find_element(By.NAME, "wyloguj").click()
# Poczekaj na przejście do strony logowania po wylogowaniu
WebDriverWait(driver, 10).until(EC.title_contains("Logowanie"))
print("Pomyślnie wylogowano")

# Zaloguj się używając danych 109266@g.elearn.uz.zgora.pl #user
driver.find_element(By.NAME, "email").send_keys("109266@g.elearn.uz.zgora.pl")
driver.find_element(By.NAME, "haslo").send_keys("123")
driver.find_element(By.CLASS_NAME, "btn-login").click()
# Poczekaj na przejście do strony użytkownika po zalogowaniu
WebDriverWait(driver, 10).until(EC.title_contains("Logowanie"))
print("Nie udało się zalogować czyli jest git bo tak miało być")
# Zaloguj się używając danych 109266@g.elearn.uz.zgora.pl #user
driver.find_element(By.NAME, "email").send_keys("109266@g.elearn.uz.zgora.pl")
driver.find_element(By.NAME, "haslo").send_keys("123")
driver.find_element(By.CLASS_NAME, "btn-login").click()

```

```
# Poczekać na przejście do strony użytkownika po zalogowaniu
WebDriverWait(driver, 10).until(
    lambda d: 'error' in parse_qs(urlparse(d.current_url).query) and
    parse_qs(urlparse(d.current_url).query)['error'][0] == '2'
)
print("Pomyślnie zablokowano użytkownika")
```

każda linijka kodu wytłumaczona w skrypcie

```
finally:

    # Zakończenie testu
    print("\nWszystkie testy zakończone")
    driver.quit()
    sys.exit()
```

zamknięcie strony i zakończenie działania skryptu wsadzone w finally po to żeby nawet jeżeli nastąpi error to zostanie wykonane

7.Wymagania systemowe

Po stronie serwera:

1. Sprzęt:

Procesor (CPU): Minimum 2 rdzenie, ale zalecane są 4 rdzenie dla lepszego zarządzania wieloma procesami, szczególnie jeśli planujesz obsługiwać większą liczbę jednoczesnych użytkowników.

Pamięć RAM: Minimum 4 GB RAM, ale zalecane 8 GB lub więcej, aby zapewnić płynne działanie MySQL i Apache, które mogą być dość zasobożerne, szczególnie przy dużych bazach danych i równoczesnych połączeniach.

Dysk twardy: Minimum 20 GB wolnego miejsca na dysku, ale zaleca się SSD dla szybszego dostępu do danych i lepszej wydajności ogólnej. Rozmiar zależy od spodziewanej ilości danych.

Łącze internetowe: Zalecana jest stabilna i szybka łączność z Internetem. Prędkość zależy od przewidywanego ruchu, ale minimum 100 Mbps symetryczne jest zalecane dla publicznych serwerów produkcyjnych.

2. Oprogramowanie:

System operacyjny: Ubuntu 22.04 LTS Server, który jest stabilną wersją z długoterminowym wsparciem, idealny do zastosowań serwerowych.

Serwer WWW: Apache 2.4.52, popularny serwer internetowy, który jest dobrze wspierany i powszechnie używany.

Baza danych: MySQL 8.0.36, silnik baz danych SQL, który wymaga odpowiedniej konfiguracji i optymalizacji w zależności od rozmiaru i charakteru danych.

PHP: Wersja 8.1.2, współczesna wersja PHP, która oferuje znaczące usprawnienia w wydajności i bezpieczeństwie w porównaniu do starszych wersji.

PHPMailer: Wersja 6.9.1, biblioteka do obsługi wysyłki e-mail z PHP, zapewniająca łatwe integracje z różnymi metodami transportu e-mail.

3. Bezpieczeństwo:

Firewall: Konfiguracja zapor ogniowych i zabezpieczeń sieciowych, aby ograniczyć nieautoryzowany dostęp do serwera.

Aktualizacje: Regularne aktualizacje oprogramowania i systemu operacyjnego w celu zapewnienia ochrony przed znanymi podatnościami.

Kopie zapasowe: Systematyczne tworzenie kopii zapasowych danych, aby zapobiec utracie danych w przypadku awarii sprzętu lub ataków.

4. Monitoring:

Narzędzia do monitorowania: Instalacja narzędzi do monitorowania zasobów systemowych, takich jak Nagios, Zabbix, czy Prometheus, aby na bieżąco śledzić wydajność i dostępność serwera.

5. Optymalizacja:

Konfiguracja MySQL: Dostosowanie ustawień MySQL, takich jak rozmiar bufora, pojemność połączeń i inne kluczowe parametry, w zależności od rzeczywistego obciążenia i użycia.

Konfiguracja Apache: Optymalizacja Apache poprzez dostosowanie liczby wątków pracowników, maksymalnej liczby połączeń i innych parametrów konfiguracji.

Przy odpowiednim konfigurowaniu, serwer spełniający te wymagania będzie w stanie obsłużyć umiarkowaną do dużą ilość ruchu, przy jednoczesnym zachowaniu dobrej wydajności i stabilności.

Po stronie użytkownika:

1. Sprzęt:

Procesor (CPU): Nie ma szczególnych wymagań, każdy współczesny procesor zapewni wystarczającą moc obliczeniową do przeglądania większości stron internetowych.

Pamięć RAM: Zaleca się co najmniej 2 GB RAM, ale 4 GB lub więcej zapewni płynniejszą obsługę, szczególnie przy otwartych wielu kartach w przeglądarce lub przy używaniu innych programów w tle.

Dysk twardy: Nie ma specjalnych wymagań co do przestrzeni dyskowej, chyba że użytkownik planuje pobierać duże ilości danych ze strony.

2. Oprogramowanie:

System operacyjny: Strona powinna być dostępna na każdym współczesnym systemie operacyjnym, w tym Windows, macOS, Linux, Android, iOS i innych, które mogą obsługiwać nowoczesne przeglądarki internetowe.

Przeglądarka internetowa: Wymagana jest przeglądarka obsługująca HTML5, CSS3 i JavaScript. Zalecane przeglądarki to najnowsze wersje Google Chrome, Mozilla Firefox, Safari, Microsoft Edge, Opera lub inne nowoczesne przeglądarki.

JavaScript: Włączone wsparcie dla JavaScript, co jest standardem w nowoczesnych przeglądarkach internetowych.

Cookies i przechowywanie lokalne: W zależności od funkcjonalności strony może być wymagane włączenie obsługi cookies oraz lokalnego przechowywania danych.

3. Łącze internetowe:

Prędkość połączenia: Prędkość połączenia zależy od rodzaju zawartości na stronie. Dla podstawowych stron internetowych wystarczające może być nawet wolniejsze połączenie DSL (np. 512 kbps). Dla stron bogatych w multimedia (np. wideo, duże obrazy, interaktywne funkcje) zalecane są szybsze łącza, np. łącze szerokopasmowe 5 Mbps lub szybsze.

4. Bezpieczeństwo:

Ochrona antywirusowa: Zaleca się korzystanie z oprogramowania antywirusowego, aby chronić urządzenie przed potencjalnie szkodliwymi plikami, które mogą być pobierane podczas przeglądania internetu.

Aktualizacje oprogramowania: Regularne aktualizacje przeglądarek internetowych i systemu operacyjnego pomagają w ochronie przed znanymi zagrożeniami i eksploatacjami bezpieczeństwa.

8.Licencja

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below,

refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this

License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or

otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions

either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively

convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

9.Autorzy

Jakub Martynański (scrummaster)

Kacper Misiaszek

Diana Yarosh

Wojciech Turlewicz