

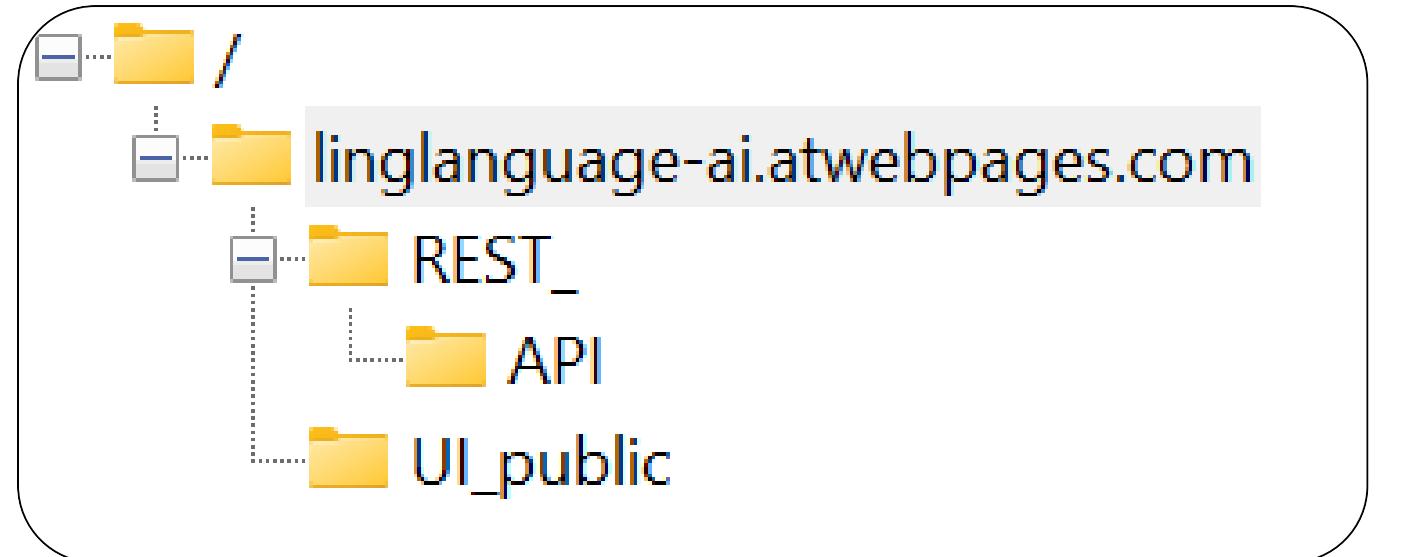


LINGLEARNAIWEB

Inteligentna aplikacja do nauki słówek z
personalizacją via AI

Struktura projektu

(Struktura folderów i plików)



UI_public/

- add_flashcard.php
- flashcards.php
- index.php
- leaderboard.html
- logout.php
- user.php
- userlogin.html
- userregister.html

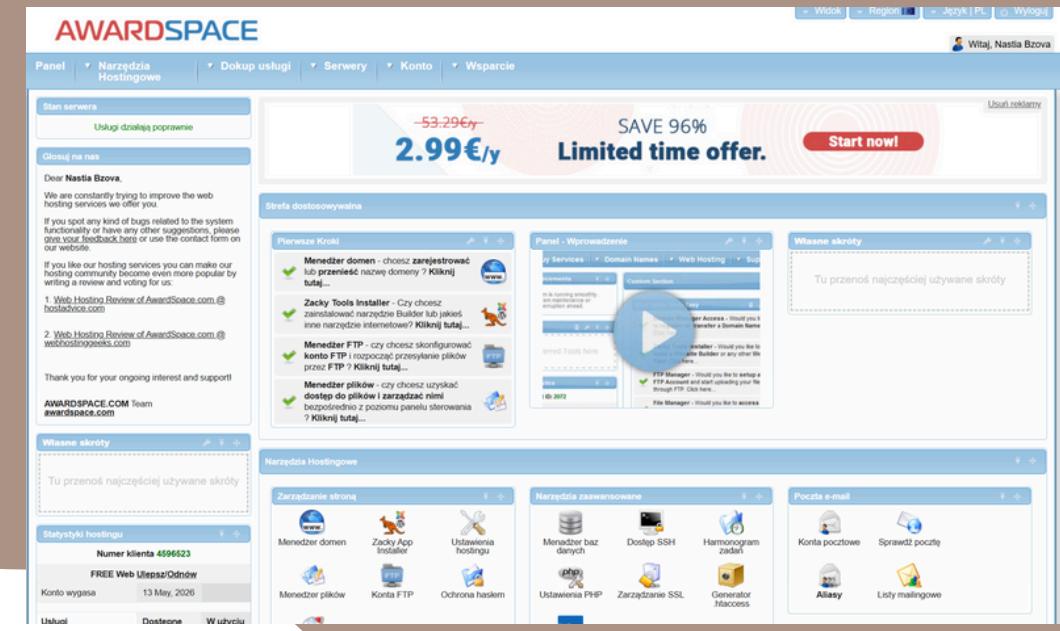
REST_ /

- ..
- API
- DB_connect.php

API/

- auth.php
- flashcards_api.php
- gamification_api.php

Struktura projektu
zamieszczonego na
AWARDSPACE.COM



Historyjki

Historyjka 1 - Nauka słówek (fiszki)

Jako zalogowany użytkownik, chcę przeglądać fiszki, aby uczyć się nowych słówek.

Kryteria akceptacji:

- Może kliknąć "Znam" lub "Nie znam".
- Wyświetlanie tłumaczenia - kiedy użytkownik nie zna słowa
- Odpowiedź użytkownika jest zapisywana w bazie.

Historyjka 2 - Logowanie i Rejestracja

Jako nowy użytkownik, chcę się zarejestrować na platformie, aby mieć dostęp do nauki fiszek.

Kryteria akceptacji:

- Formularz rejestracji zawiera pola: nazwa użytkownika, email, hasło.
- Dane są wysyłane do backendu i zapisywane w bazie.
- Użytkownik po rejestracji zostaje zalogowany i przekierowany do strony głównej.

Historyjka 3 - Gamifikacja (punkty i leaderboard)

Jako użytkownik, chcę zdobywać punkty za poprawne odpowiedzi i widzieć ranking, aby zwiększyć moją motywację do nauki.

Kryteria akceptacji:

- System punktów (np. 10 pkt za „znam”).
- Tablica wyników (leaderboard).
- Przechowywanie punktów w bazie danych.
- Wyświetlanie odznak (np. "początkujący", "mistrz" ..).

Historyjka 4 - Dodawanie własnych fiszek (wolontariuszy)

Jako wolontariusz lub administrator, chcę mieć możliwość dodawania nowych fiszek, aby wzbogacić bazę danych słówek dla innych użytkowników.

Kryteria akceptacji:

- Formularz dodawania fiszki (słówko, tłumaczenie, przykład).
- Zapis do bazy danych.
- Widoczność nowej fiszki dla innych użytkowników.

Jira

[https://scrumzadanie17.atlassian.net/jira/software/projects/LN/boards/34?
atlOrigin=eyJpljoiYzhIOTY4NTVhZTQwNDVkZWI0NTNmZjk0ZDhmNjgwOWEiLCJwljoiaiJ9](https://scrumzadanie17.atlassian.net/jira/software/projects/LN/boards/34?atlOrigin=eyJpljoiYzhIOTY4NTVhZTQwNDVkZWI0NTNmZjk0ZDhmNjgwOWEiLCJwljoiaiJ9)

Projekty
LingLanguageAlweb ...

Podsumowanie Oś czasu Backlog Tablica Kalendarz Lista Formularze Cele Wszystkie zgłoszenia

Wyszukaj na osi czasu AB Epik Kategoria statusu

Sprinty

LN-23 Implementacja funkcjonalu
LN-27 Rozwóz systemu

+ Utwórz Epik

April May June

LingLanguageAlweb SPRINT

Projekty
LingLanguageAlweb ...

Podsumowanie Oś czasu Backlog Tablica Kalendarz Lista Formularze Cele Wszystkie zgłoszenia Kod Zarchiwizowane zgłoszenia

Filtruj

Ukończono 17 w ciągu ostatnich 7 dni

Zaktualizowano 26 w ciągu ostatnich 7 dni

Utworzono 26 w ciągu ostatnich 7 dni

0 z terminem wkrótce w ciągu najbliższych 7 dni

Przegląd statusu
Get a snapshot of the status of your work items. [View all work items](#)

26 Całkowita liczba ...

Do zrobienia: 6
W toku: 3
Gotowe: 17

Ostatnia aktywność
Bądź na bieżąco z tym, co dzieje się w całym projekcie.

Dzisiaj

AB Użytkownik **Anastasiia Bzova** zaktualizował pole „description” w zgłoszeniu [LN-29: Stworzenie interfejsu użytkownika DO ZROBIENIA](#) około godziny temu

AB Użytkownik **Anastasiia Bzova** zaktualizował pole „IssueParentAssociation” w zgłoszeniu [LN-29: Stworzenie interfejsu użytkownika DO ZROBIENIA](#) około godziny temu

AB Użytkownik **Anastasiia Bzova** utworzył [LN-29: Stworzenie interfejsu użytkownika DO ZROBIENIA](#)

Nie ma jeszcze aktywności

Anastasiia Bzova abzova1@stu.vistula.edu.pl

Zakończ sp

Profile Zarządzaj ustawieniami konta Projekty

Theme Open Quickstart

Przełącz konto Wyloguj

Projekty
LingLanguageAlweb ...

Podsumowanie Oś czasu Backlog Tablica Kalendarz Lista Formularze Cele Wszystkie zgłoszenia Kod Zarchiwizowane zgłoszenia

Wyszukaj na tablicy AB Epik

DO ZROBIENIA 1
Implementacja algorytmu SM-2
ROZWÓJ SYSTEMU
LN-24

W TOKU 1 MAKS.: 2
Nauka słówek (fiszki)
IMPLEMENTACJA FUNKCJONALU
LN-8

GOTOWE 3
Logowanie i Rejestracja
IMPLEMENTACJA FUNKCJONALU
LN-1

Gamifikacja (punkty i leaderboard)
IMPLEMENTACJA FUNKCJONALU
LN-12

Dodawanie własnych fiszek (wolontariuszy)
IMPLEMENTACJA FUNKCJONALU
LN-17

+ Utwórz

Zakończ sp

Projekty
LingLanguageAlweb ...

Podsumowanie Oś czasu Backlog Tablica Kalendarz Lista Formularze Cele Wszystkie zgłoszenia Kod Zarchiwizowane zgłoszenia

Wyszukaj na liście AB Filtruj

	Typ	Klucz	Podsumowanie	Status	Komentarze	Sprint	Osoba przypisana
<input type="checkbox"/>		LN-23	Implementacja funkcjonalu	W TOKU	<input type="button" value="Dodaj komentarz"/>		
<input type="checkbox"/>		LN-1	Logowanie i Rejestracja	GOTOWE	<input type="button" value="Dodaj komentarz"/>	LingLanguageAlweb	
<input type="checkbox"/>		LN-4	Formularz rejestracji projektu UI	GOTOWE	<input type="button" value="Dodaj komentarz"/>	LingLanguageAlweb	
<input type="checkbox"/>		LN-5	Wdrożenie front-endu formularza rejestracyjnego	GOTOWE	<input type="button" value="Dodaj komentarz"/>	LingLanguageAlweb	
<input type="checkbox"/>		LN-6	Opracowanie interfejsu API zaplecza do rejestracji	GOTOWE	<input type="button" value="Dodaj komentarz"/>	LingLanguageAlweb	
<input type="checkbox"/>		LN-7	Wdrożenie logowania użytkownika po rejestracji	GOTOWE	<input type="button" value="Dodaj komentarz"/>	LingLanguageAlweb	
<input type="checkbox"/>		LN-12	Gamifikacja (punkty i leaderboard)	GOTOWE	<input type="button" value="Dodaj komentarz"/>	LingLanguageAlweb	
<input type="checkbox"/>		LN-17	Dodawanie własnych fiszek (wolontariuszy)	GOTOWE	<input type="button" value="Dodaj komentarz"/>	LingLanguageAlweb	
<input type="checkbox"/>		LN-8	Nauka słówek (fiszki)	W TOKU	<input type="button" value="Dodaj komentarz"/>	LingLanguageAlweb	
<input type="checkbox"/>		LN-27	Rozwój systemu	DO ZROBIENIA	<input type="button" value="Dodaj komentarz"/>		
<input type="checkbox"/>		LN-24	Implementacja algorytmu SM-2	DO ZROBIENIA	<input type="button" value="Dodaj komentarz"/>	LingLanguageAlweb	
<input type="checkbox"/>		LN-26	Zintegruj algorytm SM-2 z aplikacją	DO ZROBIENIA	<input type="button" value="Dodaj komentarz"/>		
<input type="checkbox"/>		LN-28	Ocena trudności karty	DO ZROBIENIA	<input type="button" value="Dodaj komentarz"/>	LingLanguageAlweb	
<input type="checkbox"/>		LN-29	Stworzenie interfejsu użytkownika	DO ZROBIENIA	<input type="button" value="Dodaj komentarz"/>	LingLanguageAlweb	

Commits

Search commits master [Show all](#)

Author	Commit	Message	Time Ago
Anastasia Bzova	fd01f4b	flashcards_api.php edited - added add_flashcard action	
Anastasia Bzova	38a9d83	MERGED Merged in UI_public (pull request #4) UI public	
Anastasia Bzova	2db8db9	MERGED Merged in REST_API (pull request #3) REST API	
Anastasia Bzova	e8c7e52	user.php created profile for user	
Anastasia Bzova	ca84292	add_flashcard.php created to addd flashcards	
Anastasia Bzova	bbec767	UI_public/index.html deleted	15 minutes ago
Anastasia Bzova	8241808	index.html renamed into index.php and added new logic "add_flashcards"	15 minutes ago
Anastasia Bzova	9685f05	MERGED Merged in LoginRegisterUser (pull request #2) LoginRegisterUser	17 minutes ago
Anastasia Bzova	ecdadd5	flashcards.php created	19 minutes ago
Anastasia Bzova	5cbb3c1	logout.php created to end user session	
Anastasia Bzova	4e18051	auth.php created API for authorization user in system	
Anastasia Bzova	9358fd0	userregister.html created Register user to system	
Anastasia Bzova	f4ccf8e	userlogin.html created Login user to system	
Anastasia Bzova	52b4a67	leaderboard.html created top users	
Anastasia Bzova	31baa6b	index.html created	
Anastasia Bzova	aaafaaa7	gamification_api.php created	
Anastasia Bzova	50d5e62	flashcards_api.php created	
Anastasia Bzova	2d03ffb	MERGED Merged in REST_API (pull request #1) REST API	
Anastasia Bzova	a2ecbff	DB_connect.php edited online with Bitbucket	
Anastasia Bzova	48d96da	DB_connect.php connection to DataBase	51 minutes ago
Anastasia Bzova	ba86c65	DB_connect.php connection to DataBase	1 hour ago
Anastasia Bzova	5cabb11	Initial commit	2 hours ago

BitBucket

Account

Anastasiia Bzova
abzova@stu.vistula.edu.pl[Switch account](#)Theme [BETA](#)

Recent workspaces



All workspaces

Log out

15 minutes ago

15 minutes ago

1 15 minutes ago

17 minutes ago

19 minutes ago

myownworkspace / LingLanguageAlweb

LingLanguageAlweb

Here's where you'll find this repository's source files. To give your users an idea of what they'll find here, [add a description to your repository](#).

master	Files	Filter files	<input type="text"/>
/			
Name	Size	Last commit	Message
REST_		9 minutes ago	flashcards_api.php edited - added add_flashcard action
UI_public		13 minutes ago	user.php created profile for user
.gitignore	624 B	2 hours ago	Initial commit

UI design strony

UI bazowana na modernym [Bootstrap](#) framework CSS, za pomocą którego można zadbać o wizualną stronę witryn internetowych oraz aplikacji.

The login screen features a large orange 'Witamy w LingLanguageAI' header with a hand icon. Below it is a sub-header 'Zaloguj się lub zarejestruj, aby rozpocząć naukę!'. It includes three buttons: 'Zaloguj się' (blue), 'Zarejestruj się' (grey), and 'Ranking' (green).

The welcome screen shows a large orange 'Witamy w LingLanguageAI' header with a hand icon. Below it is a greeting 'Witaj, Nastia:3!' and a sub-header 'Wybierz jedną z opcji, aby rozpocząć swoją własną naukę!'. It includes three buttons: 'Ranking' (green), 'Nauka fiszek' (blue), and 'Mój profil' (cyan).

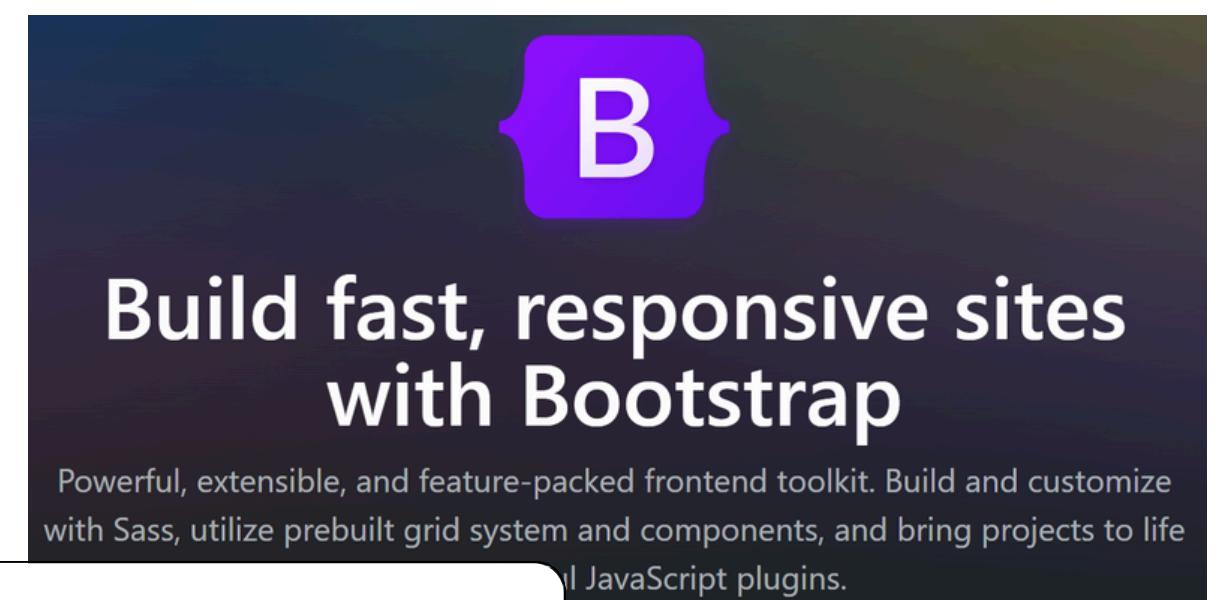
The ranking screen has a green header 'Ranking użytkowników' with a trophy icon. It displays a table of users with their names and scores:

#	Użytkownik	Punkty
1	language_lover	500
2	jan_kowalski	440
3	Test123456789	90
4	anna_nowak	75
5	Nastia:3	70
6	പ്രമ്മിഗ്ഷി	20
7	test_user	0

A blue button at the bottom left says '< Powrót do menu głównego'.

The top part shows a 'Nauka fiszek' card for the word 'joy' with 'Znam' checked. A 'Skończyć sesję' button is below it. To the right is a 'Mój profil' card for user 'Nastia:3' showing stats: Punkty: 70, Odznaki: beginner. Below is a 'Stwórz fiszki i podziel się z resztą!' section with a 'Dodaj +' button. A 'Wyloguj się' button is in the top right corner.

The bottom part shows a 'Dodaj nową fiszkę' form with fields for 'Słówko' (fire), 'Tłumaczenie' (ogień), and 'Przykładowe zdanie' (Shiny fire). It includes a 'Zapisz fiszkę' button and a '← Powrót' button.



Baza danych

(Tworzenie tabel w bazie danych oraz ich struktura)

Users Uruchom zapytanie SQL/zapytania w tabeli 4596523_linglanguageai.Users: [?](#)

```
1 CREATE TABLE Users (
2     id INT AUTO_INCREMENT PRIMARY KEY,
3     username VARCHAR(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
4     email VARCHAR(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci,
5     password VARCHAR(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
6     points INT DEFAULT 0,
7     badges TEXT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci,
8     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
9     role ENUM('user', 'volunteer') CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci DEFAULT 'user'
10 );
11 
```

//Uzytkowniki

Flashcards

Uruchom zapytanie SQL/zapytania w tabeli 4596523_linglanguageai.Users: [?](#)

```
1 CREATE TABLE Flashcards (
2     id INT AUTO_INCREMENT PRIMARY KEY,
3     word VARCHAR(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
4     translation VARCHAR(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci,
5     difficulty ENUM('łatwe', 'średnie', 'trudne') CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci DEFAULT 'średnie',
6     created_by INT,
7     created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
8     example_sentence TEXT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL
9 );
10 
```

//Fiszki

phpMyAdmin

Ostatnie Favorites

- 4596523_linglanguageai
 - + New
 - + Flashcards
 - + FlashcardSessions
 - + Users

FlashcardSessions Uruchom zapytanie SQL/zapytania w tabeli 4596523_linglanguageai.Users: [?](#)

```
1 CREATE TABLE FlashcardSessions (
2     id INT AUTO_INCREMENT PRIMARY KEY,
3     user_id INT,
4     flashcard_id INT,
5     result ENUM('znam', 'nie_znam') CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci,
6     review_date DATE
7 );
8 
```

//Sesji nauki użytkownika

Wypełnienie tablic danymi

Uruchom zapytanie SQL/zapytania w tabeli 4596523_linglanguageai.Flashcards:

```
1 INSERT INTO Flashcards (id, word, translation, difficulty) VALUES
2 (1, 'apple', 'jabłko', 'łatwe'),
3 (2, 'computer', 'komputer', 'łatwe'),
4 (3, 'perseverance', 'wytrwałość', 'trudne'),
5 (4, 'sunshine', 'światło słoneczne', 'średnie'),
6 (5, 'bicycle', 'rower', 'łatwe'),
7 (6, 'serendipity', 'szczęśliwy traf', 'trudne'),
8 (7, 'watermelon', 'arbuz', 'średnie'),
9 (8, 'enthusiasm', 'entuzjazm', 'średnie'),
10 (9, 'book', 'książka', 'łatwe'),
11 (10, 'magnificent', 'wspaniały', 'trudne'),
12 (11, 'dog', 'pies', 'łatwe'),
13 (12, 'cat', 'kot', 'łatwe'),
14 (13, 'ephemeral', 'efemeryczny', 'trudne'),
15 (14, 'delicious', 'pyszny', 'średnie'),
16 (15, 'ocean', 'ocean', 'łatwe'),
17 (16, 'butterfly', 'motyl', 'łatwe'),
18 (17, 'complicated', 'skomplikowany', 'trudne'),
19 (18, 'garden', 'ogród', 'łatwe'),
20 (19, 'phenomenon', 'zjawisko', 'średnie'),
21 (20, 'joy', 'radość', 'łatwe');
```

Dodawanie przykładowych danych
do istniejących tablic

Uruchom zapytanie SQL/zapytania w tabeli 4596523_linglanguageai.Users:

```
1 INSERT INTO Users (id, username, email, points, badges, created_at) VALUES
2 (1, 'jan_kowalski', 'jan.kowalski@example.com', 120, 'beginner,fast_learner', '2025-01-15 10:30:00'),
3 (2, 'anna_nowak', 'anna.nowak@mail.com', 75, 'beginner', '2025-02-20 14:15:00'),
4 (3, 'test_user', 'test.user@test.com', 250, 'beginner,fast_learner,advanced', '2025-03-10 08:45:00'),
5 (4, 'language_lover', 'lang.lover@domain.com', 500, 'fast_learner,advanced,expert', '2025-05-05 18:20:00');
```

flashcards_api.php

Implementacja

AWARDSPACE

Panel | Narzędzia Hostingowe | Dokup usługi | Serwery | Konto | Wsparcie

Menedżer plików

- linglanguage-ai.atwebpages.com
- REST_
- API
- UI_public

Wróć | Close | Cofnij zmiany | Zapisz

/home/www/linglanguage-ai.atwebpages.com/REST_/AF | Poziom wyżej | Dom

```
<?php
session_start();
require_once '../DB_connect.php';
header('Content-Type: application/json');

$method = $_SERVER['REQUEST_METHOD'];

if ($method === 'GET' && $_GET['action'] === 'get_flashcards') {
    // Pobierz losowe fiszki
    $result = $MySQLconection->query("SELECT * FROM Flashcards ORDER BY RAND() LIMIT 5");

    $flashcards = [];
    while ($row = $result->fetch_assoc()) {
        $flashcards[] = $row;
    }

    echo json_encode($flashcards);
    exit;
}

elseif ($method === 'POST' && $_GET['action'] === 'submit_answer') {
    // Odczytaj dane z JSON-a
    $data = json_decode(file_get_contents("php://input"), true);
    $userId = $data['user_id'];
    $flashcardId = $data['flashcard_id'];
    $result = $data['result'];

    // Zapisanie wyników sesji nauki użytkownika
    $stmt = $MySQLconection->prepare("INSERT INTO FlashcardSessions (user_id, flashcard_id, result, review_date) VALUES (?, ?, ?, CURDATE())");
    $stmt->bind_param("iis", $userId, $flashcardId, $result);
    $stmt->execute();

    // Aktualizacja punktów (10 pkt za poprawną odpowiedź)
    if ($result === 'znam') {
        $stmt2 = $MySQLconection->prepare("UPDATE Users SET points = points + 10 WHERE id = ?");
        $stmt2->bind_param("i", $userId);
        $stmt2->execute();
    }

    echo json_encode(["success" => true]);
    exit;
}

elseif ($method === 'POST' && $_GET['action'] === 'add_flashcard') {
    $data = json_decode(file_get_contents('php://input'), true);
    $word = $data['word'] ?? '';
    $translation = $data['translation'] ?? '';
    $example = $data['example_sentence'] ?? '';

    $example = $data['example_sentence'] ?? '';

    if ($word && $translation && $example) {
        $stmt = $MySQLconection->prepare("INSERT INTO Flashcards (word, translation, example_sentence, created_by) VALUES (?, ?, ?, ?)");
        $stmt->bind_param("ssss", $word, $translation, $example, $_SESSION['user']['id']);
        if ($stmt->execute()) {
            echo json_encode(['status' => 'success']);
        } else {
            http_response_code(500);
            echo json_encode(['status' => 'error', 'message' => 'Błąd zapisu']);
        }
    } else {
        http_response_code(400);
        echo json_encode(['status' => 'error', 'message' => 'Brak danych']);
    }
}

else {
    http_response_code(400);
    echo json_encode(["error" => "Nieprawidłowe żądanie"]);
}
?>
```

Jesteś nowy w naszym menedżerze plików?
Odkryj wszystkie wbudowane funkcje

Otwórz Samouczek

flashcards_api.php X

C: > Users > Nastia > AppData > Local > Temp > fz3temp-2 > flashcards_api.php

```
<?php
session_start();
require_once '../DB_connect.php';
header('Content-Type: application/json');

$method = $_SERVER['REQUEST_METHOD'];

if ($method === 'GET' && $_GET['action'] === 'get_flashcards') {
    // Pobierz losowe fiszki
    $result = $MySQLconection->query("SELECT * FROM Flashcards ORDER BY RAND() LIMIT 5");

    $flashcards = [];
    while ($row = $result->fetch_assoc()) {
        $flashcards[] = $row;
    }

    echo json_encode($flashcards);
    exit;
}

elseif ($method === 'POST' && $_GET['action'] === 'submit_answer') {
    // Odczytaj dane z JSON-a
    $data = json_decode(file_get_contents("php://input"), true);
    $userId = $data['user_id'];
    $flashcardId = $data['flashcard_id'];
    $result = $data['result'];

    // Zapisanie wyników sesji nauki użytkownika
    $stmt = $MySQLconection->prepare("INSERT INTO FlashcardSessions (user_id, flashcard_id, result, review_date) VALUES (?, ?, ?, CURDATE())");
    $stmt->bind_param("iis", $userId, $flashcardId, $result);
    $stmt->execute();

    // Aktualizacja punktów (10 pkt za poprawną odpowiedź)
    if ($result === 'znam') {
        $stmt2 = $MySQLconection->prepare("UPDATE Users SET points = points + 10 WHERE id = ?");
        $stmt2->bind_param("i", $userId);
        $stmt2->execute();
    }

    echo json_encode(["success" => true]);
    exit;
}

elseif ($method === 'POST' && $_GET['action'] === 'add_flashcard') {
    $data = json_decode(file_get_contents('php://input'), true);
    $word = $data['word'] ?? '';
    $translation = $data['translation'] ?? '';
    $example = $data['example_sentence'] ?? '';

    $example = $data['example_sentence'] ?? '';

    if ($word && $translation && $example) {
        $stmt = $MySQLconection->prepare("INSERT INTO Flashcards (word, translation, example_sentence, created_by) VALUES (?, ?, ?, ?)");
        $stmt->bind_param("ssss", $word, $translation, $example, $_SESSION['user']['id']);
        if ($stmt->execute()) {
            echo json_encode(['status' => 'success']);
        } else {
            http_response_code(500);
            echo json_encode(['status' => 'error', 'message' => 'Błąd zapisu']);
        }
    } else {
        http_response_code(400);
        echo json_encode(['status' => 'error', 'message' => 'Brak danych']);
    }
}

else {
    http_response_code(400);
    echo json_encode(["error" => "Nieprawidłowe żądanie"]);
}
?>
```

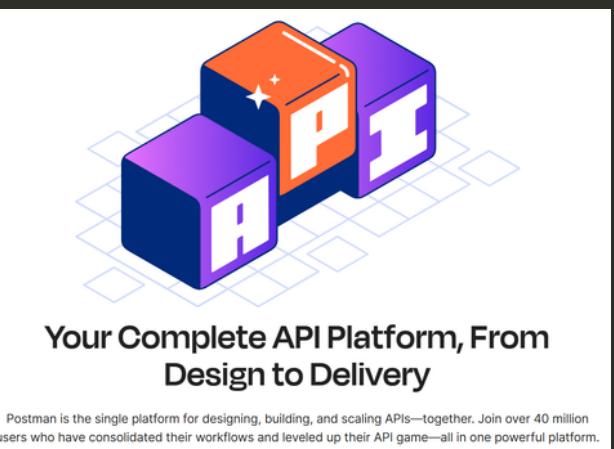
Testowanie - flashcards_api

(Za pomocą Postman).

GET: http://linglanguage-ai.atwebpages.com/REST_API/flashcards_api.php?action=get_flashcards

wyniki:

```
[{"id": "8", "word": "enthusiasm", "translation": "entuzjazm", "difficulty": "\u015brednie"}, {"id": "20", "word": "joy", "translation": "rado\u015b\u0107", "difficulty": "\u0142atwe"}, {"id": "18", "word": "garden", "translation": "ogr\u00f3d", "difficulty": "\u0142atwe"}, {"id": "7", "word": "watermelon", "translation": "arbuz", "difficulty": "\u015brednie"}, {"id": "14", "word": "delicious", "translation": "pyszny", "difficulty": "\u015brednie"}]
```



POST: http://linglanguage-ai.atwebpages.com/REST_API/flashcards_api.php?action=submit_answer

HTTP http://linglanguage-ai.atwebpages.com/REST_API/flashcards_api.php?action=submit_answer

POST http://linglanguage-ai.atwebpages.com/REST_API/flashcards_api.php?action=submit_answer

Params • Authorization Headers (9) Body • Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "user_id": 1,  
3   "flashcard_id": 4,  
4   "result": "znam"  
5 }  
6
```

wyniki:

Body Cookies (1) Headers (6) Test Results

Pokazano wiersze 0 - 0 (1 total, Wykonanie zapytania trwało 0.0147 sekund(y)).

SELECT * FROM `FlashcardSessions`

{ } JSON ▾ ▷ Preview ⏪ Visualize ▾

	id	user_id	flashcard_id	result	review_date
1	2	1	4	znam	2025-05-14

POST: http://linglanguage-ai.atwebpages.com/REST_API/flashcards_api.php?action=add_flashcard

HTTP http://linglanguage-ai.atwebpages.com/REST_API/flashcards_api.php?action=add_flashcard

POST http://linglanguage-ai.atwebpages.com/REST_API/flashcards_api.php?action=add_flashcard

Params • Authorization Headers (9) Body • Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "word": "banana",  
3   "translation": "banan",  
4   "example_sentence": "He peeled the banana."  
5 }
```

wyniki:

Body Cookies (1) Headers (9) Test Results

{ } JSON ▾ ▷ Preview ⏪ Visualize ▾

	id	word	translation	difficulty	example_sentence	created_by	created_at
1	13	ephemeral	efemeryczny	trudne	He peeled the banana.	NULL	2025-05-15 20:21:44
2	30	banana	banan	średnie	He peeled the banana.	NULL	2025-05-15 21:09:48

gamification_api.php

Implementacja

The screenshot shows the AwardSpace interface. On the left, there's a sidebar with links like 'Panel', 'Narzędzia Hostingowe', 'Dokup usługi', 'Serwery', 'Konto', and 'Wsparcie'. Below that is the 'Menedżer plików' (File Manager) for the domain 'linglanguage-ai.atwebpages.com'. It lists three folders: 'REST_', 'API', and 'UI_public'. The 'REST_' folder is selected, and its contents are displayed in the main editor area. The editor shows PHP code for the 'gamification_api.php' file. The code handles two main actions: 'get_leaderboard' (returning the top 10 users by points) and 'get_user_gamification' (returning user points and badges). It uses MySQL queries to fetch data and JSON encoding to return the results. The editor has tabs for 'Wróć', 'Close', 'Cofnij zmiany', and 'Zapisz'. Below the editor is a preview pane showing the URL '/home/www/linglanguage-ai.atwebpages.com/REST_/AF' and a 'Poziom wyżej' (Higher Level) button. A sidebar on the right contains a message about being new to the file manager and a 'Samouczek' (Tutorial) button.

```
gamification_api.php X leaderboard.html

gamification_api.php
1 <?php
2 require_once '../DB_connect.php';
3
4 header('Content-Type: application/json');
5
6 $method = $_SERVER['REQUEST_METHOD'];
7
8 if ($method === 'GET' && $_GET['action'] === 'get_leaderboard') {
9     // Pobierz ranking użytkowników (top 10)
10    $result = $MySQLconection->query("SELECT id, username, points FROM Users ORDER BY points DESC LIMIT 10");
11
12    $leaderboard = [];
13    while ($row = $result->fetch_assoc()) {
14        $leaderboard[] = $row;
15    }
16
17    echo json_encode($leaderboard);
18
19} elseif ($method === 'GET' && $_GET['action'] === 'get_user_gamification' && isset($_GET['user_id'])) {
20    // Pobierz punkty i odznaki danego użytkownika z jednej tabeli
21    $userId = intval($_GET['user_id']);
22
23    $stmt = $MySQLconection->prepare("SELECT username, points, badges FROM Users WHERE id = ?");
24    $stmt->bind_param("i", $userId);
25    $stmt->execute();
26    $result = $stmt->get_result();
27    $data = $result->fetch_assoc();
28
29    if ($data) {
30        // Zamień string odznak na tablicę
31        $badges = array_map('trim', explode(',', $data['badges']));
32
33        echo json_encode([
34            "username" => $data['username'],
35            "points" => intval($data['points']),
36            "badges" => $badges
37        ]);
38    } else {
39        http_response_code(404);
40        echo json_encode(["error" => "Użytkownik nie znaleziony"]);
41    }
42
43} else {
44    http_response_code(400);
45    echo json_encode(["error" => "Nieprawidłowe żądanie"]);
46}
47 ?>
```

Testowanie - *gamification_api*

GET:

http://linglanguage-ai.atwebpages.com/REST_API/gamification_api.php?action=get_leaderboard

wyniki:

```
[{"id": "4", "username": "language_lover", "points": "500"}, {"id": "3", "username": "test_user", "points": "250"}, {"id": "1", "username": "jan_kowalski", "points": "130"}, {"id": "2", "username": "anna_nowak", "points": "75"}]
```

POST:

Otrzymanie info o
użytkowniku >

http://linglanguage-ai.atwebpages.com/REST_API/gamification_api.php?action=get_user_gamification&user_id=1

wynik:

```
{"username": "jan_kowalski", "points": 440, "badges": ["beginner", "fast_learner"]}
```

http://linglanguage-ai.atwebpages.com/REST_API/gamification_api.php?action=get_user_gamification&user_id=2

wynik:

```
{"username": "anna_nowak", "points": 75, "badges": ["beginner"]}
```

Otrzymanie
leaderboard >

http://linglanguage-ai.atwebpages.com/REST_API/gamification_api.php?action=get_leaderboard

wynik:

```
[{"id": "4", "username": "language_lover", "points": "500"}, {"id": "3", "username": "test_user", "points": "250"}, {"id": "1", "username": "jan_kowalski", "points": "130"}, {"id": "2", "username": "anna_nowak", "points": "75"}]
```

auth_api.php

Implementacja

AWARDSPACE

Panel | Narzędzia Hostingowe

Dokup usługi

Serwery

Konto

Wsparcie

Menedżer plików

linglanguage-ai.atwebpages.com

REST_

API

UI_public

Wróć Close Cofnij zmiany Zapisz

/home/www/linglanguage-ai.atwebpages.com/REST_AF Poziom wyżej Dom

```
1 <?php
2 require_once '../DB_connect.php';
3 session_start();
4 header('Content-Type: application/json');
5
6 $method = $_SERVER['REQUEST_METHOD'];
7 $action = $_GET['action'] ?? '';
8
9 if ($method === 'POST' && $action === 'register') {
10    // Rejestracja użytkownika
11    $data = json_decode(file_get_contents("php://input"), true);
12    $username = $data['username'];
13    $email = $data['email'];
14    $password = password_hash($data['password'], PASSWORD_BCRYPT);
15
16    // Sprawdź czy email już istnieje
17    $stmt = $MySQLconection->prepare("SELECT id FROM Users WHERE email = ?");
18    $stmt->bind_param("s", $email);
19    $stmt->execute();
20    $stmt->store_result();
21
22    if ($stmt->num_rows > 0) {
23        echo json_encode(["success" => false, "error" => "Email już istnieje."]);
24        exit;
25    }
26
27    $stmt->close();
28
29    // Dodaj użytkownika
30    $created_at = date('Y-m-d H:i:s');
31    $badges = "beginner";
32    $points = 0;
33
34    $stmt = $MySQLconection->prepare("INSERT INTO Users (username, email, password, points, badges, created_at) VALUES (?, ?, ?, ?, ?, ?)");
35    $stmt->bind_param("ssssss", $username, $email, $password, $points, $badges, $created_at);
36
37    if ($stmt->execute()) {
38        $userId = $stmt->insert_id;
39
40        $_SESSION['user'] = [
41            "id" => $userId,
42            "username" => $username
43        ];
44
45        echo json_encode([
46            "success" => true,
47            "user" => [
48                "id" => $userId,
49                "username" => $username
50            ]
51        ]);
52    } else {
53        echo json_encode(["success" => false, "error" => "Błąd"]);
54    }
55
56    $stmt->close();
57
58 } elseif ($method === 'POST' && $action === 'login') {
59
60    // Logowanie użytkownika
61    $data = json_decode(file_get_contents("php://input"), true);
62    $email = $data['email'];
63    $password = $data['password'];
64
65    $stmt = $MySQLconection->prepare("SELECT id, username, password FROM Users WHERE email = ?");
66    $stmt->bind_param("s", $email);
67    $stmt->execute();
68    $stmt->store_result();
69
70    if ($stmt->num_rows === 0) {
71        echo json_encode(["success" => false, "error" => "Nieprawidłowy email lub hasło."]);
72        exit;
73    }
74
75    $stmt->bind_result($id, $username, $hashedPassword);
76    $stmt->fetch();
77
78    // Sprawdź hasło
79    if (password_verify($password, $hashedPassword)) {
80
81        $_SESSION['user'] = [
82            "id" => $id,
83            "username" => $username
84        ];
85
86        echo json_encode([
87            "success" => true,
88            "user" => [
89                "id" => $id,
90                "username" => $username
91            ]
92        ]);
93
94    } else {
95        echo json_encode(["success" => false, "error" => "Nieprawidłowe hasło."]);
96    }
97
98    $stmt->close();
99
100 } else {
101     http_response_code(400);
102     echo json_encode(["error" => "Niepoprawne zapytanie."]);
103 }
```

Jesteś nowy w naszym menedżerze plików?



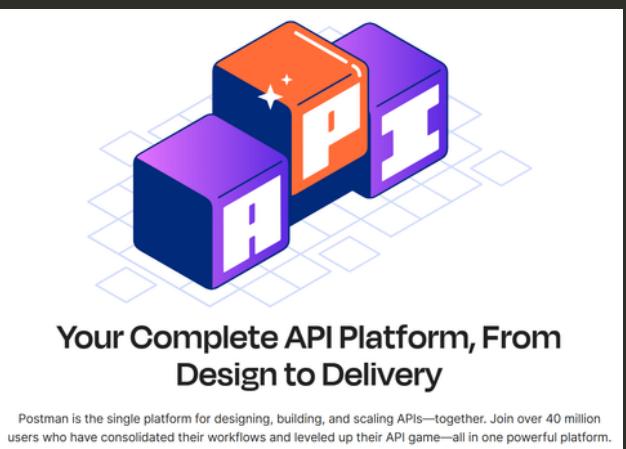
Odkryj wszystkie wbudowane funkcje

Otwórz Samouczek

```
auth.php x
C: > Users > Nastia > AppData > Local > Temp > fz3temp-2 > auth.php
9 if ($method === 'POST' && $action === 'register') {
10    // Rejestracja użytkownika
11    $data = json_decode(file_get_contents("php://input"), true);
12    $username = $data['username'];
13    $email = $data['email'];
14    $password = password_hash($data['password'], PASSWORD_BCRYPT);
15
16    // Sprawdź czy email już istnieje
17    $stmt = $MySQLconection->prepare("SELECT id FROM Users WHERE email = ?");
18    $stmt->bind_param("s", $email);
19    $stmt->execute();
20    $stmt->store_result();
21
22    if ($stmt->num_rows > 0) {
23        echo json_encode(["success" => false, "error" => "Email już istnieje."]);
24        exit;
25    }
26
27    $stmt->close();
28
29    // Dodaj użytkownika
30    $created_at = date('Y-m-d H:i:s');
31    $badges = "beginner";
32    $points = 0;
33
34    $stmt = $MySQLconection->prepare("INSERT INTO Users (username, email, password, points, badges, created_at) VALUES (?, ?, ?, ?, ?, ?)");
35    $stmt->bind_param("ssssss", $username, $email, $password, $points, $badges, $created_at);
36
37    if ($stmt->execute()) {
38        $userId = $stmt->insert_id;
39
40        $_SESSION['user'] = [
41            "id" => $userId,
42            "username" => $username
43        ];
44
45        echo json_encode([
46            "success" => true,
47            "user" => [
48                "id" => $userId,
49                "username" => $username
50            ]
51        ]);
52    } else {
53        echo json_encode(["success" => false, "error" => "Błąd"]);
54    }
55
56    $stmt->close();
57
58 } elseif ($method === 'POST' && $action === 'login') {
59
60    // Logowanie użytkownika
61    $data = json_decode(file_get_contents("php://input"), true);
62    $email = $data['email'];
63    $password = $data['password'];
64
65    $stmt = $MySQLconection->prepare("SELECT id, username, password FROM Users WHERE email = ?");
66    $stmt->bind_param("s", $email);
67    $stmt->execute();
68    $stmt->store_result();
69
70    if ($stmt->num_rows === 0) {
71        echo json_encode(["success" => false, "error" => "Nieprawidłowy email lub hasło."]);
72        exit;
73    }
74
75    $stmt->bind_result($id, $username, $hashedPassword);
76    $stmt->fetch();
77
78    // Sprawdź hasło
79    if (password_verify($password, $hashedPassword)) {
80
81        $_SESSION['user'] = [
82            "id" => $id,
83            "username" => $username
84        ];
85
86        echo json_encode([
87            "success" => true,
88            "user" => [
89                "id" => $id,
90                "username" => $username
91            ]
92        ]);
93
94    } else {
95        echo json_encode(["success" => false, "error" => "Nieprawidłowe hasło."]);
96    }
97
98    $stmt->close();
99
100 } else {
101     http_response_code(400);
102     echo json_encode(["error" => "Niepoprawne zapytanie."]);
103 }
```

Testowanie - auth_api

(Za pomocą [Postman](#)).



POST: http://linglanguage-ai.atwebpages.com/REST_API/auth.php?action=register

HTTP http://linglanguage-ai.atwebpages.com/REST_API/auth.php?action=register

POST http://linglanguage-ai.atwebpages.com/REST_API/auth.php?action=register

Params • Authorization Headers (8) Body • Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "username": "test_user",  
3   "email": "test@mail.com",  
4   "password": "123456"  
5 }  
6
```

wynik:

Body Cookies Headers (6) Test Results

{ } JSON ▾ Preview Visualize

```
1 {  
2   "success": true,  
3   "user": {  
4     "id": 7,  
5     "username": "test_user"  
6   }  
7 }
```

POST: http://linglanguage-ai.atwebpages.com/REST_API/auth.php?action=login

HTTP http://linglanguage-ai.atwebpages.com/REST_API/auth.php?action=login

POST http://linglanguage-ai.atwebpages.com/REST_API/auth.php?action=login

Params • Authorization Headers (8) Body • Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "email": "test@mail.com",  
3   "password": "1234567"  
4 }  
5
```

prawidłowe
hasło

wyniki:

Body Cookies Headers (6) Test Results

{ } JSON ▾ Preview Visualize

```
1 {  
2   "success": false,  
3   "error": "Nieprawidłowy email lub hasło."  
4 }
```

prawidłowe
hasło

NIE prawidłowe
hasło

HTTP http://linglanguage-ai.atwebpages.com/REST_API/auth.php?action=login

POST http://linglanguage-ai.atwebpages.com/REST_API/auth.php?action=login

Params • Authorization Headers (8) Body • Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "email": "test@mail.com",  
3   "password": "123456"  
4 }  
5
```

NIE prawidłowe
hasło

Body Cookies Headers (6) Test Results

{ } JSON ▾ Preview Visualize

```
1 {  
2   "success": true,  
3   "user": {  
4     "id": 7,  
5     "username": "test_user"  
6   }  
7 }
```

**ZAPRASZAM
ODWIEDZIĆ STRONĘ**

http://linglanguage-ai.atwebpages.com/UI_public/

DZIĘKUJĘ ZA UWAGĘ



SCAN ME