



# פרויקט ORT: מערכת Q&A חכמה לכיתה

מערכת שאלות ותשובות מתקדמת המשלבת בינה מלאכותית לשיפור חווית הלמידה. המערכת מיועדת למורים וסטודנטים, ומציעה חיפוש סמנטי חכם וניתוח תשובות אוטומטי.

## תכונות מרכזיות

- ניהול שאלות כיתתיות ותשובות סטודנטים בצורה יעילה ומאורגנת
- תשובות תלמידים: שליחה וניהול של תשובות תלמידים
- בקרת גישה: אימות מורים מבוסס קוד גישה
- חיפוש חכם: חיפוש סמנטי באמצעות שאלות שפה טבעית
- סיכום בינה מלאכותית: ניתוח תשובות תלמידים באמצעות הוראות מותאמות אישית

# ארכיטקטורה

## תיאור התרשים:

**צד לקוח:** אפליקציית React/TypeScript שמציגה את הממשק למורה ולתלמיד.

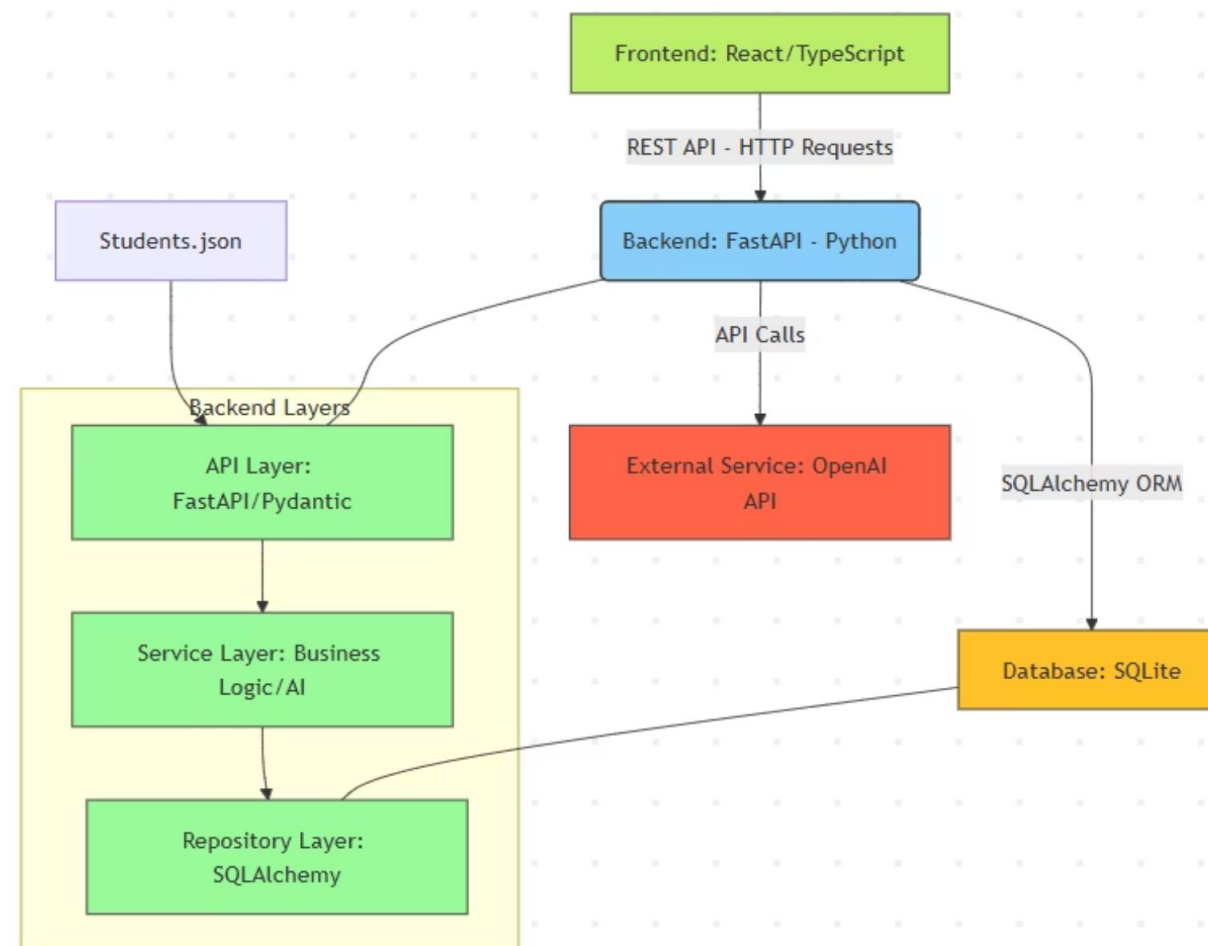
**תקשורת (REST API):** ה-Frontend שולח בקשות HTTP ל-Backend.

**צד שרת:** יישום FastAPI ב-Python המטפל בלוגיקה העסקית.

**שכבות פנימיות (Layers):** מציג את חלוקת האחריות הפנימית של ה-Backend ל-API, Service ו-Repository.

של ה-Backend ל-API, Service ו-Repository.

**שירות חיצוני:** שירות OpenAI API לביצוע Smart Search ו-AI Summarization.



# ספריות ושירותים

## לקוח - Frontend

### React 18

**ספריית UI:** ממשק משתמש מודרני מבוסס קומפוננטות פונקציונליות ו-Hooks

### TypeScript

**בטיחות טיפוסים:** קוד בטוח יותר, הפחתת שגיאות ושיפור חווית פיתוח

### Tailwind CSS

פיתוח מהיר של עיצוב רספונסיבי: **CSS Utility-First**

### Vite

**Build Tool:** HMR כלי בנייה מהיר עם

## שרת - Backend

### FastAPI

**שרת API:** פיתוח מהיר של REST API עם ביצועים גבוהים (ASGI/Uvicorn) ומסמכי API אוטומטיים (Swagger UI)

### Pydantic

**וולידציה:** בדיקת תקינות נתונים נכנסים ויוצאים, כולל נתוני AI

### SQLAlchemy

בצורה אובייקטית SQLite ניהול אינטראקציה עם **ORM:**

### Uvicorn

**שרת ASGI:** שרת אסינכרוני עם תמיכה

### pytest

בדיקה: סיפרייה לצורכי טסטים לקוד

### Openai

ספרייה לעבודה נוחה עם המודל chatGPT

# כלים נילווים לתמיכה ופיתוח

הכלים הבאים תומכים ומבטיחים פיתוח, בדיקה ופריסה חלקה של הפרויקט, תוך שמירה על איכות קוד גבוהה ואוטומציה מלאה.



## Docker

**קונטיינריזציה:** סביבת פיתוח אחידה עם  
docker-compose.yml המאפשרת הרצה  
מיידיית של Backend ו-Frontend יחד



## DBeaver

**ניהול נתונים:** כלי GUI אינטואיטיבי לבדיקה,  
הרצת שאילתות וניפוי שגיאות של SQLite



## Cursor

**פיתוח תוכנה:** כלי לפיתוח תוכנה IDE המשלב  
בתוכו יכולות פיתוח AI בעזרת צ'ט שיועד לקרוא  
context שך הקוד. יצירת קובץ cursor rule  
ליצירת חוקים עבור LLM לפיתוח אחיד וחלק יותר.



## Gemini

**ייעוץ והכוונה:** LLM לצורכי ייעוץ ומענה לשאלות  
במהלך הפיתוח.



## GitHub Actions

הפעלת בדיקות אוטומטיות על כל **CI/CD**  
עם בדיקות מטריצה, Pull Request ו-Commit  
Python ו-Node.js לגרסאות



## Postman

**בדיקות תקשורת:** כלי המיועד לבדיקות ה-API  
של השרת.



## pytest

**בדיקות שרת:** Framework לבדיקות יחידה  
(Unit) ואינטגרציה (Integration) של שירותי  
השאלות, התשובות וה-API.

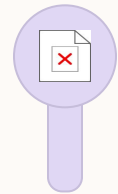


## Vitest

**בדיקות לקוח:** Framework מהיר לבדיקות  
קומפוננטות React, Hooks וזרימות משתמש  
(User Workflows).

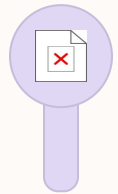
# מבנה Backend בארכיטקטורה שכבתית

השרת בנוי בארכיטקטורה שכבתית המבטיחה הפרדת דאגות ובדיקותיות גבוהה:



## API Layer

ניתוב קריאות - POST /api/v1/questions/open



## Service Layer

לוגיקה עסקית - QuestionService, AnswerService



## Repository Layer

גישה לבסיס נתונים - QuestionRepository

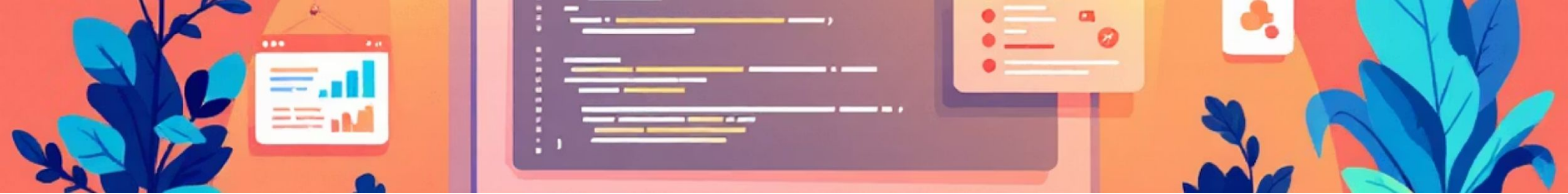


## Data Layer

מודלים - Question Model, Answer Model

```
backend/
├── app/                                # Main application code
│   ├── __init__.py                    # Package initialization
│   └── main.py                         # FastAPI application
├── entry point
│   └── api/                            # API layer
│       └── api.py                      # Main API router
├── configuration
│   ├── endpoints/                     # API endpoint modules
│   ├── config/                        # Configuration modules
│   ├── database/                      # Database layer
│   └── config.py                      # Database configuration
├── and connection
│   ├── models/                        # SQLAlchemy ORM models
│   ├── repositories/                 # Data access layer
│   ├── models/                       # Pydantic models for API
│   ├── services/                     # Business logic layer
│   └── utils/                         # Utility functions
├── tests/                             # Test suite
├── requirements.txt                   # Python dependencies
├── pytest.ini                         # pytest configuration
├── run_tests.py                       # Test runner script
├── set_db_path.py                     # Database path
├── configuration
├── database_schema.sql                # Database schema
├── definition
├── Dockerfile                         # Docker configuration
├── .dockerignore                      # Docker ignore file
├── .gitignore                         # Git ignore file
└── README.md                          # Backend documentation
```





# מבנה Frontend

מבנה קבצים מודולרי המבטיח קריאות, ארגון וקלות תחזוקה. כל רכיב במערכת ממוקם בתיקייה ייעודית עם הפרדה ברורה בין שרת ולקוח.

## מבנה התיקיות הראשי

```
ort-frontend/src/
├── components/           # Reusable UI
components
├── pages/               # Page components
├── hooks/               # Custom React hooks
├── types/               # TypeScript type
definitions
├── theme.css            # Custom Tailwind CSS
theme
├── App.tsx              # Main application
component
├── main.tsx             # Application entry
point
```



### פיתוח

שימוש ב-**TypeScript** ו-**Vite** לשמירה על Type Safety ובניית פרויקט מהירה.



### עיצוב ו-Styling

שימוש ב-**Tailwind CSS** ליצירת עיצוב Responsive (Mobile-first) ומהיר, עם דגש על נגישות (**Accessibility**).



### טכנולוגיות

React 18 מותאמים אישית Hooks עם Custom Hooks API ואחסור נתונים מה State לניהול.

# תמונות מהפרויקט

## תצוגה של שאלה

עמוד התצוגה של השאלה המציג את השאלה תשובות התלמידים וסיכום AI

### Question Details

#### Geography class

Open Access Code: G001  
2 answers (22%) Created: Oct 18, 2025, 04:59 PM

#### Question Prompt

What is capital of france?

Close Question Summary with AI

#### Student Responses

2 Responses

| STUDENT                        | ANSWER TEXT | SUBMISSION TIME      |
|--------------------------------|-------------|----------------------|
| Niv Greenberg<br>ID: STU1002   | london      | 18.10.2025, 17:01:32 |
| Shalom Hamelech<br>ID: STU1004 | paris       | 18.10.2025, 17:01:06 |

#### AI Summary

Clear Summary

One student, Shalom Hamelech, correctly identified Paris as the capital of France. However, another student, Niv Greenberg, provided an incorrect answer by stating London as the capital.

### Teacher Dashboard

Manage questions and view student answers

Create Question Logout

Total Questions

2

Open Questions

2

Closed Questions

0

Questions

Filter by Status

All QuestionsOpen QuestionsClosed Questions

Search Questions

Smart Search (e.g., "geography", "math problems", "science questions", "history")

Type your search query...Search

Try searching for:

general knowledgegeographymathematicssciencehistoryliteraturecapital citiesmath problemsscience questions

Geography class

What is capital of france?

G0012 answers (22%)Created: Oct 18, 2025, 04:59 PMOpen

View Details

Sports

Best Sport ever?

S0011 answer (11%)Created: Oct 18, 2025, 05:00 PMOpen

View Details

ORT Assignment System

Teacher Dashboard

Student Form

### Student Answer Form

Enter your access code and student ID to answer a question.

#### Answer Question

##### Geography class

What is capital of france?

Access Code: G001Open

Your Answer

paris

5/200 characters

Back to Access Code

Submit Answer

## דשבורד מורה

הצגה של כל השאלות פתוחות וסגורות, עם אופציה של יצירת

שאלה, חיפוש חכם של שאלות וסינון בן שאלות פתוחות או סגורות

## תשובת תלמיד

טופס הגשת תשובה של תלמיד שדורשת היזדהות קודם לכן של

מספר תלמיד ומספר שאלה.במידה שהשאלה סגורה הטופס לא

יישלח. ולא ניתנת אפשרות של הכנסה של יותר מ200 תוים.

Made with GAMMA

# בסיס הנתונים - SQLite

בסיס נתונים קל משקל מבוסס דיסקת, פשוט להטמעה, ללא צורך בהגדרת שרת נתונים נפרד, אידיאלי לסביבת פיתוח. עבור בדיקות, נעשה שימוש ב-**in-memory SQLite** מבודד עבור כל הרצת בדיקה, מה שמבטיח נתונים נקיים ועצמאיות בין הבדיקות.

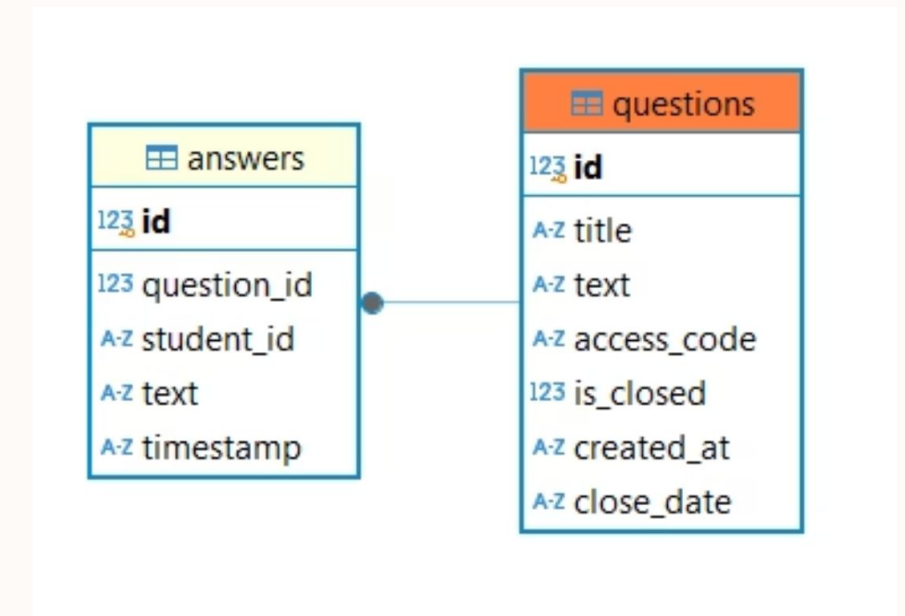
## גישה לנתונים (Data Access) – שכבת ה-ORM

י- **Object-Relational Mapper (ORM)** - SQLAlchemy: ספריית ה-ORM שנבחרה. מאפשרת עבודה עם בסיס הנתונים באמצעות אובייקטי Python, במקום כתיבת שאילתות SQL גולמיות.

י- Data Models - מוגדרים מודלים כגון **Question Model** ו-**Answer Model**.

י- **Repository Layer** - הגישה לבסיס הנתונים נעשית דרך **שכבת ה-Repository**, שמטרתה להפריד את לוגיקת הגישה לנתונים מהלוגיקה של הסרוויס (Service Layer).

גרף ERD המציין קשר של אחד לרבים בין התשובות לשאלות



## כלי עזר לניהול הנתונים: DBeaver

כלי ניהול: **DBeaver** מומלץ ככלי ניהול (GUI) עוצמתי וחופשי.

פונקציונליות: מאפשר למפתחים לחקור טבלאות, להריץ שאילתות SQL מותאמות אישית ולנפות שגיאות הקשורות לבסיס הנתונים SQLite באופן אינטואיטיבי.



כלי AI בפיתוח – ניתוח, אינטגרציה ותובנות מרכזיות

| תובנות ושיטות עבודה מומלצות   | אתגרים מרכזיים  | שימוש/תפקיד עיקרי   | כלי AI                             |
|---|---|---|------------------------------------|
| <b>MCP (Model Context Protocol):</b><br>Context7-Postman-שימוש ב<br>לשיפור הדיוק על ידי הרחבת הקונטקסט<br>(תיעוד, API). | מגבלות חלון ההקשר: צורך בבחירה<br>סלקטיבית של קבצים לקריאה.           | הבנה וניתוח קוד: ניתוח בסיס קוד קיים,<br>הבנת זרימה וזיהוי דפוסים ארכיטקטוניים.     | 1. Cursor AI (כלי פיתוח עיקרי)     |
| חיוני לשמירה על<br>סטנדרטים ארכיטקטוניים וסגנון קוד   | דיוק ההיסק: לעיתים נדרשה קריאה חוזרת<br>להבנה מלאה.                   | שימוש מתקדם: אכיפת סגנון ועקביות קוד.   |                                    |
| שימוש <b>JSON Response Format:</b><br>response_format: {"type":<br>הבטיח אמינות גבוהה {"json_object":<br>בפלט.          | הנדסת פרומפטים: דרישה לניסוח קפדני<br>של System Prompts לקבלת עקביות. | שירותי AI מובנים: שירות סיכום (לתשובות<br>סטודנטים) ושירות חיפוש סמנטי.             | 2. OpenAI API (Backend אינטגרציית) |
| כוונון טמפרטורה: נמוכה (0.3) לפלט<br>מובנה; גבוהה (0.7) לסיכומים יצירתיים.  | טיפול בשגיאות: API failures, Rate<br>Limiting ותגובות לא תקינות.      | פורמט פלט עקבי: הבטחת פלט JSON<br>עקבי לעיבוד נתונים.                               |                                    |
| חוזקות משלימות: כל כלי מצטיין בתחום<br>אחר (קוד, אינטגרציה, ארכיטקטורה).  | מעבר הקשר: קושי בשמירה על עקביות<br>ההבנה במעבר בין כלי AI שונים.     | ראייה ארכיטקטונית: ניתוח מבנה מערכת<br>כללי, אימות דפוסי עיצוב (Best<br>Practices). | 3. Gemini (ייעוץ וניתוח)           |