## ✅ Authentication API – /api/auth/

This section handles user authentication including registration, login, and password management.

Note: The endpoints here are included from users.urls, so the exact implementation depends on your project.

Common headers:

Content-Type: application/json

Accept: application/json

---

### 🔑 POST /api/auth/register/

Description: Register a new user account.

Request body example:

```
{
  "username": "johndoe",
  "email": "john@example.com",
  "password": "strongpassword123"
}
```

Response example:

```
{
  "id": 1,
  "username": "johndoe",
  "email": "john@example.com"
}
```

---

### 🔑 POST /api/auth/login/

Description: Authenticate a user and retrieve access token/session.

Request body example:

```
{
  "username": "johndoe",
  "password": "strongpassword123"
}
```

Response example:

```
{
  "token": "your_auth_token_here",
  "user": {
    "id": 1,
    "username": "johndoe"
  }
}
```

---

🔑 POST /api/auth/logout/

Description: Log out the current user and invalidate the session/token.

Request: No body needed.

Response example:

```
{
  "message": "Logged out successfully."
}
```

---

🔑 POST /api/auth/password_reset/

Description: Request a password reset email.

Request body example:

```
{
  "email": "john@example.com"
}
```

Response example:

```
{
  "message": "Password reset link sent to email."
}
```

---

## 📗 Flashcards API (Level 1) – /api/level1/flashcards/

This API is responsible for managing flashcards that help students in learning.

---

### ✅ GET /api/level1/flashcards/

Description: Retrieve a list of flashcards.

Response example:

```
[
  {
    "id": 1,
    "question": "What is a variable?",
    "answer": "A variable stores data values.",
    "tags": ["programming", "basics"],
    "difficulty": "easy"
  },
  {
    "id": 2,
    "question": "Explain recursion.",
    "answer": "A function calling itself...",
    "tags": ["advanced"],
    "difficulty": "medium"
  }
]
```

---

✅ POST /api/level1/flashcards/

Description: Create a new flashcard.

Request body example:

```
{
  "question": "What is inheritance?",
  "answer": "A feature that allows one class to derive from another.",
  "tags": ["OOP"],
  "difficulty": "medium"
}
```

Response example:

```
{
  "id": 3,
  "question": "What is inheritance?",
  "answer": "A feature that allows one class to derive from another.",
  "tags": ["OOP"],
  "difficulty": "medium"
}
```

---

✅ GET /api/level1/flashcards/{id}/

Description: Retrieve details of a specific flashcard.

Response example:

```
{
  "id": 3,
  "question": "What is inheritance?",
  "answer": "A feature that allows one class to derive from another.",
  "tags": ["OOP"],
  "difficulty": "medium"
}
```

---

✅ PUT /api/level1/flashcards/{id}/

Description: Update an existing flashcard.

Request body example:

```
{
  "question": "What is inheritance in OOP?",
  "answer": "A mechanism where one class inherits from another.",
  "tags": ["OOP", "concept"],
  "difficulty": "medium"
}
```

Response example:

```
{
  "id": 3,
  "question": "What is inheritance in OOP?",
  "answer": "A mechanism where one class inherits from another.",
  "tags": ["OOP", "concept"],
  "difficulty": "medium"
}
```

---

✅ DELETE /api/level1/flashcards/{id}/

Description: Delete a flashcard.

Response example:

```
{
  "message": "Flashcard deleted successfully."
}
```

---

## 📘 Documents API (Level 1) – /api/level1/documents/

This API manages documents that are resources for generating flashcards and notes.

---

### ✅ GET /api/level1/documents/

Description: Retrieve all uploaded documents.

Response example:

```
[
  {
    "id": 1,
    "title": "Math Notes",
    "file_url": "http://<domain>/media/documents/math_notes.pdf"
  }
]
```

---

### ✅ POST /api/level1/documents/

Description: Upload a new document.

Request (multipart/form-data):

title: "Physics Notes"

file: <file upload>

Response example:

```
{
  "id": 2,
  "title": "Physics Notes",
  "file_url": "http://<domain>/media/documents/physics_notes.pdf"
}
```

---

✅ GET /api/level1/documents/{id}/

Description: Retrieve details of a specific document.

Response example:

```
{
  "id": 2,
  "title": "Physics Notes",
  "file_url": "http://<domain>/media/documents/physics_notes.pdf"
}
```

---

✅ PUT /api/level1/documents/{id}/

Description: Update document details.

Request body example:

```
{
  "title": "Updated Physics Notes"
}
```

Response example:

```
{
  "id": 2,
  "title": "Updated Physics Notes",
  "file_url": "http://<domain>/media/documents/physics_notes.pdf"
}
```

---

✅ DELETE /api/level1/documents/{id}/

Description: Delete a document.

Response example:

```
{
  "message": "Document deleted successfully."
}
```

---

📘 Quizzes API (Level 2) – /api/level2/quizzes/

This API manages quizzes designed for student assessment.

---

✅ GET /api/level2/quizzes/

Description: Retrieve a list of quizzes.

Response example:

```
[
  {
    "id": 1,
    "title": "Math Quiz 1",
    "questions_count": 10,
    "duration_minutes": 30
  }
]
```

---

✅ POST /api/level2/quizzes/

Description: Create a new quiz.

Request body example:

```
{
  "title": "Science Quiz 2",
  "questions_count": 15,
  "duration_minutes": 45
}
```

Response example:

```
{
  "id": 2,
  "title": "Science Quiz 2",
  "questions_count": 15,
  "duration_minutes": 45
}
```

---

✅ GET /api/level2/quizzes/{id}/

Description: Retrieve details of a specific quiz.

Response example:

```
{
  "id": 2,
  "title": "Science Quiz 2",
  "questions_count": 15,
  "duration_minutes": 45
}
```

---

✅ PUT /api/level2/quizzes/{id}/

Description: Update a quiz.

Request body example:

```
{
  "title": "Updated Science Quiz 2",
  "questions_count": 20,
  "duration_minutes": 50
}
```

Response example:

```
{
  "id": 2,
  "title": "Updated Science Quiz 2",
  "questions_count": 20,
  "duration_minutes": 50
}
```

---

✅ DELETE /api/level2/quizzes/{id}/

Description: Delete a quiz.

Response example:

```
{
  "message": "Quiz deleted successfully."
}
```

---

📊 Dashboard API – /api/dashboard/stats/

✅ GET /api/dashboard/stats/

Description: Retrieve statistics such as the total number of documents, quizzes, flashcards, and user activity.

Response example:

```
{
  "total_documents": 12,
  "total_flashcards": 48,
  "total_quizzes": 8,
  "user_activity": {
    "today": 4,
    "this_week": 20
  }
}
```

---

## 📘 Notes Generator API (Level 3) – /api/level3/

This API allows users to upload documents and generate notes from them.

---

### ✅ POST /api/level3/upload/

Description: Upload a document for note generation.

Request (multipart/form-data):

file: <file upload>

Response example:

```
{
  "file_id": 5,
  "file_url": "http://<domain>/media/uploads/notes.pdf",
  "message": "File uploaded successfully."
}
```

---

### ✅ GET /api/level3/generate-notes/

Description: Generate notes based on the uploaded file.

Query parameters:

- file_id: The ID of the uploaded file (optional if handled differently).

Response example:

```
{
  "notes": [
    {
      "question": "Define Newton's First Law.",
      "answer": "An object at rest stays at rest..."
    },
    {
      "question": "Explain acceleration.",
      "answer": "The rate of change of velocity..."
    }
  ]
}
```

---

📂 Media Files in Development

If DEBUG = True in your settings, files uploaded through the APIs will be accessible at:

http://<domain>/media/<file_path>

For example:
http://127.0.0.1:8000/media/uploads/notes.pdf

---

✅ Summary Table

| Endpoint | Method | Description |
|---|---|---|
| /api/auth/register/ | POST | Register a new user |
| /api/auth/login/ | POST | Login and get authentication |
| /api/auth/logout/ | POST | Logout |
| /api/auth/password_reset/ | POST | Reset password request |
| /api/level1/documents/ | GET | List documents |
| /api/level1/documents/ | POST | Upload a document |
| /api/level1/documents/{id}/ | GET | Get document details |
| /api/level1/documents/{id}/ | PUT | Update document |
| /api/level1/documents/{id}/ | DELETE | Delete document |
| /api/level1/flashcards/ | GET | List flashcards |
| /api/level1/flashcards/ | POST | Create flashcard |
| /api/level1/flashcards/{id}/ | GET | Get flashcard details |
| /api/level1/flashcards/{id}/ | PUT | Update flashcard |
| /api/level1/flashcards/{id}/ | DELETE | Delete flashcard |
| /api/level2/quizzes/ | GET | List quizzes |
| /api/level2/quizzes/ | POST | Create quiz |
| /api/level2/quizzes/{id}/ | GET | Get quiz details |
| /api/level2/quizzes/{id}/ | PUT | Update quiz |
| /api/level2/quizzes/{id}/ | DELETE | Delete quiz |
| /api/dashboard/stats/ | GET | Get dashboard statistics |
| /api/level3/upload/ | POST | Upload file for notes generation |
| /api/level3/generate-notes/ | GET | Generate notes from uploaded file |