**EduNotes AI: Technology Stack Breakdown**

**FRONTEND TECHNOLOGIES**

1. React

- What: JavaScript library for building dynamic UIs.

- Why Used: Enables role-specific dashboards (Student, Teacher, Admin) with reusable components.

- Importance: Perfect for handling real-time note previews, AI summaries, and interactive features.

- Suitability: Integrates seamlessly with Django REST APIs.

2. Tailwind CSS

- What: Utility-first CSS framework.

- Why Used: Rapidly design responsive interfaces (e.g., note cards, upload modals).

- Importance: Saves time with pre-built classes; ensures consistency across dashboards.

- Suitability: Ideal for custom designs without heavy CSS.

3. Axios

- What: HTTP client for API calls.

- Why Used: Fetch notes, submit uploads, and handle JWT authentication.

- Importance: Simplifies communication between React and Django backend.

- Suitability: Lightweight and supports async/await.

4. React Router

- What: Navigation library for React.

- Why Used: Route users to pages like `/dashboard`, `/upload`, `/notes/:id`.

- Importance: Manages role-based access (e.g., Teachers see moderation tools).

- Suitability: Essential for SPAs with multiple views.

**BACKEND TECHNOLOGIES**

5. Django

- What: Python web framework.

- Why Used: Handle user auth, note management, and AI pipelines.

- Importance: Built-in admin panel for moderating notes/users.

- Suitability: Robust security and scalability for academic data.

6. Django REST Framework (DRF)

- What: Toolkit for building REST APIs in Django.

- Why Used: Expose endpoints for notes, summaries, and recommendations.

- Importance: Decouples frontend/backend; supports JWT auth.

- Suitability: Perfect for React + Django stacks.

7. PostgreSQL

- What: Relational database.

- Why Used: Store structured data (users, notes, courses, universities).

- Importance: Handles complex queries (e.g., "Show all CSE notes from MIT").

- Suitability: Django’s ORM works flawlessly with PostgreSQL.

8. JWT (JSON Web Tokens)

- What: Token-based authentication.

- Why Used: Secure login for Students/Teachers/Admins.

- Importance: Stateless and scalable for distributed systems.

- Suitability: Standard for modern web apps.

**AI TECHNOLOGIES**

9. OpenAI API (GPT-3.5/4)

- What: LLM for text processing.

- Why Used: Generate note summaries and answer subject queries (chatbot).

- Importance: Enhances usability with smart previews.

- Suitability: Easy integration via Python SDK.

10. Tesseract OCR

- What: Optical Character Recognition engine.

- Why Used: Extract text from handwritten notes/images.

- Importance: Makes non-digital notes searchable.

- Suitability: Open-source and works with Django.

11. Scikit-learn / Sentence Transformers

- What: ML libraries for embeddings/similarity.

- Why Used: Power recommendation system ("Similar Notes").

- Importance: Personalizes content discovery.

- Suitability: Lightweight for Django integration.

12. Joblib/Pickle

- What: Model serialization tools.

- Why Used: Save trained recommendation models.

- Importance: Avoid retraining on every request.

- Suitability: Works with scikit-learn.

**VISUALIZATION & UTILITIES**

13. Recharts

- What: React charting library.

- Why Used: Display analytics (e.g., "Trending Notes," user engagement).

- Importance: Helps admins/teachers track platform usage.

- Suitability: Simple and customizable.

14. Firebase Storage/AWS S3

- What: Cloud storage solutions.

- Why Used: Store PDFs, images, and processed notes.

- Importance: Scalable and secure file handling.

- Suitability: Offloads server load; integrates with Django.

15. Django Channels (Optional)

- What: WebSockets for real-time features.

- Why Used: Live notifications (e.g., "Your note got 5 upvotes").

- Importance: Boosts engagement.

- Suitability: Advanced but powerful for gamification.

**Role-Based Access Summary**

| Role | Access |

|-------------|---------------------------------------------------------------------------|

| Student | Upload/view notes, rate content, get recommendations. |

| Teacher | Verify notes, moderate content, access analytics. |

| Admin | Manage users, enforce policies, monitor AI systems. |