

Let me break down the complete system architecture for your English teaching app:

1. **Frontend Layer (React/Next.js)**

* **Core Components:**
* Authentication UI (login, registration, profile)
* Lesson Interface (interactive learning environment)
* Speech Interface (real-time speech interaction)
* Analytics Dashboard (progress tracking)
* **Features:**
* Responsive design for mobile/desktop
* Real-time speech recognition/synthesis
* Interactive lesson UI
* Progress visualization

1. **Backend Services (FastAPI/Python)**

* **API Layer:**
* RESTful endpoints for all features
* WebSocket support for real-time communication
* Request validation and error handling
* **Service Modules:**
* **Auth Service (Supabase Auth)**
* User authentication and authorization
* JWT token management
* Social auth integration (if needed)
* **LLM Service (Cohere)**
* Conversation management
* Language correction
* Lesson generation
* Writing feedback
* **Speech Service (Azure)**
* Speech-to-text conversion
* Text-to-speech synthesis
* Pronunciation assessment

1. **Database Layer (Supabase)**

* **Tables:**

1. -- Users table

2. users (

3. id uuid primary key,

4. email text unique,

5. full\_name text,

6. language\_level text,

7. learning\_goals jsonb,

8. created\_at timestamp,

9. last\_login timestamp

10. )

11.

12. -- Lessons table

13. lessons (

14. id uuid primary key,

15. user\_id uuid references users(id),

16. title text,

17. content jsonb,

18. type text,

19. difficulty level,

20. created\_at timestamp

21. )

22.

23. -- Progress table

24. progress (

25. id uuid primary key,

26. user\_id uuid references users(id),

27. lesson\_id uuid references lessons(id),

28. score numeric,

29. completed\_at timestamp,

30. feedback jsonb

31. )

32.

33. -- Conversations table

34. conversations (

35. id uuid primary key,

36. user\_id uuid references users(id),

37. messages jsonb[],

38. context text,

39. created\_at timestamp

40. )

41.

42. -- Analytics table

43. analytics (

44. id uuid primary key,

45. user\_id uuid references users(id),

46. metric\_type text,

47. metric\_value jsonb,

48. recorded\_at timestamp

49. )

50.

1. **External Services Integration**

* **Cohere API Integration:**
* Chat completion endpoints
* Prompt engineering for different lesson types
* Error handling and rate limiting
* **Azure Speech Services:**
* Real-time speech recognition
* Natural text-to-speech conversion
* Accent and pronunciation analysis

1. **Security Measures**

* JWT-based authentication
* HTTPS encryption
* API rate limiting
* Input validation
* Secure storage of API keys using environment variables

1. **Deployment Architecture**

1. /backend

2. ├── api/

3. │ ├── routes/

4. │ │ ├── auth.py

5. │ │ ├── lessons.py

6. │ │ ├── speech.py

7. │ │ └── analytics.py

8. │ ├── services/

9. │ │ ├── cohere\_service.py

10. │ │ ├── speech\_service.py

11. │ │ └── supabase\_service.py

12. │ └── utils/

13. │ ├── auth.py

14. │ └── helpers.py

15. ├── tests/

16. └── config/

17.

18. /frontend

19. ├── components/

20. │ ├── auth/

21. │ ├── lessons/

22. │ ├── speech/

23. │ └── common/

24. ├── pages/

25. ├── styles/

26. └── utils/

27.

1. **Key Features to Implement:**

* Real-time conversation practice
* Automated grammar correction
* Pronunciation feedback
* Progress tracking and analytics
* Personalized lesson generation
* Speech recognition and synthesis
* User profile and progress management

1. **Scalability Considerations:**

* Implement caching for frequently accessed data
* Use connection pooling for database
* Implement rate limiting for API endpoints
* Consider serverless deployment for cost optimization