Layered Architecture

WonderNest

Layers

1. Presentation Layer
2. Business Layer
3. Persistence Layer
4. Database Layer
5. Integration Layer
6. Infrastructure Layer

Presentation Layer

Purpose: This is the user-facing layer responsible for interacting with the kid and parent. It includes the UI/UX and handles user inputs/outputs.

Components/Services:

- **Web Application:** Frontend application for kids and parents.
- **Voice Assistant:** Interface for interacting via speech.
- **Mobile App:** Touch-based UI for activities and progress tracking.

Technologies:

- **Frontend Framework:** React.js
- **Voice Assistant Framework:** Google Dialogflow or custom implementation
- **Mobile Development:** Flutter

Business Layer

Purpose: Implements the core business logic and domain rules, such as validating activities, scoring progress, and analyzing behavior.

Components/Services:

- **Activity Classes:** Flashcard, StoryGenerator, SentenceWriting, Puzzle
- **Progress Tracker:** Tracks progress and updates scores.
- **Behavioral Analysis:** Validates input and detects inappropriate behavior.
- **AIEngine:** Generates stories, answers, and grammar corrections.
- **RecommendationEngine:** Provides suggestions for the next activities.

Technologies:

- **Business Logic Implementation:** Plain JavaScript classes
- **AI Models:** PyTorch, or OpenAI API
- **Recommendation Models:** Collaborative filtering or rule-based algorithms

Persistence Layer

Purpose: Manages interactions with the databases and other storage systems. Provides an abstraction layer for CRUD operations.

Components/Services:

- **Data Access Objects (DAO):** Encapsulates all database operations for specific entities such as User, Activity, Progress, and BehaviorLog.
  - **UserDAO:** Handles user-related database operations.
  - **ActivityDAO:** Manages activity-related queries.
  - **ProgressDAO:** Retrieves progress and stores updates.
- Simplifies database queries by providing reusable methods for CRUD operations.

Technologies:

- **ORM Tools:** Mongoose (MongoDB)
- **Sequalize:** Node.js

Database Layer

Purpose: Responsible for storing and retrieving data such as user profiles, progress, activities, and generated content.

Components/Services:

- **Database:** Stores user profiles, activities, and progress data.
- **Content Storage:** Stores assets like flashcards, conventional and generated stories, and audio files.
- **Behavioral Logs:** Logs inappropriate behavior for future reference.

Technologies:

- **Database:** MongoDB (NoSQL for flexibility)
- **Cloud Storage:** Firebase Storage for media assets
- **ORM:** Mongoose

Integration Layer

Purpose: Facilitates communication with external services and APIs, such as generative AI models, cloud storage. This layer provides a unified interface for managing these integrations and abstracts away the complexities of third-party APIs.

Components/Services:

- **Generative AI Integration:** Manages interaction with third-party AI services for story generation, grammar correction, and question answering.
  - **Technology:** OpenAI API.
- **Cloud Storage Integration:** Handles the storage and retrieval of media files, such as generated stories, images, and audio files.
  - **Technology:** Azure Blob Storage.

Infrastructure Layer

Purpose: Provides the foundational support required for hosting, scalable deployment, and automated processes such as building, testing, and deployment.

Components/Services:

- **Hosting Platform:**
  - Provides the environment where the application runs.
  - Ensures scalability, fault tolerance, and availability.
  - **Technology:** Azure App Service.
- **Containerization:**
  - Ensures the application and its dependencies run consistently across different environments (development, testing, and production).
  - **Technology:** Docker for containerization, Kubernetes for orchestration.
- **CI/CD Pipeline:**
  - Automates the processes of building, testing, and deploying code to the production environment.
  - Ensures seamless integration of changes and quick rollouts.
- **Technology:** GitHub Actions.

Thank you!