OwlAI: Backend Logic Document – Home, Library, Projects & Discover Modules

---

Module 1: Home (Chatbot Interface)

Purpose:

The Home screen hosts the chatbot that acts as the student's study companion. It is the entry point for doubt-solving, practice, theory explanations, and contextual learning prompts.

Core Components of Chatbot Backend:

1. Query Logging

- Log each user input with:
    - Timestamp
    - Session ID
    - Subject & unit tag (via keyword mapping)
    - Query type: definition, theory, MCQ, example-based
    - Complexity: short / medium / long

2. Feedback & Reinforcement Capture

- Explicit: Thumbs up / down
- Implicit:
    - Follow-up within 30 seconds = confusion
    - Time spent reading response
    - Copy-clicks = interest

3. Contextual Prompt Memory (within session)

- Maintain short-term memory within the same session to handle follow-ups logically.
- e.g., "Explain more" or "Give another example" maps to previous query.

4. Streaks & Motivation Tracking

- Count queries per day
- Session durations
- Micro-achievements: "10 queries today! 🎉"

5. Session Grouping

- Start-time, end-time, queries asked, total interaction length.
- Use this for personalized session summaries later.

6. Voice Input / Hinglish Layer (Phase 2)

- Integrate ASR (Automatic Speech Recognition) tagging logic.
- Identify language preferences for future multilingual support.

Module 2: Library (Smart History & Learning Timeline)

Purpose:

To transform raw chatbot queries into a structured, Topic-wise, user-friendly knowledge base. It provides students with an intelligent view of what they've learned, how often they revisit concepts, and what they still need to revise.

Core Components:

1. Topics Covered

- Tag each query to a unit/chapter/topic.
- Track total queries, last accessed date, subtopic coverage.
- Add query type & complexity labels.

2. Time Map (Calendar Heatmap)

- Store usage like `{ "2025-04-12": 3 }`
- Bundle into study sessions (session ID).

3. Revisit Tracker

- Flag repeated/confused questions.
- Fallback: follow-ups without feedback, short dwell time.

4. Glossary Builder

- Automatically capture definition-type queries.
- Build personal dictionary.

5. Progress Tracker per Unit

- Based on query diversity, quality, feedback.

6. Weekly Summary Generator

- List of concepts explored.
- Flag weak areas.

7. Retraining Loop Prep

- Store low-quality responses flagged by user.

8. Privacy Layer

- Anonymized logs
- User opt-out option for data tracking

Module 3: Projects (Student Study Collections)

Purpose:

To allow students to curate their own saved learning material, including past questions, summaries, test items, and notes — like a study folder or Notion-lite.

Core Components:

1. Create & Manage Projects

- Project name, subject tag, topic tag
- Timestamped creation & edit logs

2. Add Content Blocks

- From chatbot: save response to project
- From student: allow notes / screenshot uploads
- Support tagging (e.g., “MCQ”, “Summary”, “PYQ”)

3. Quiz/Test Mode

- Allow user to create quiz based on saved MCQs
- Track scores per project
- Show improvement over time

4. Auto-Summarize (Future)

- Backend identifies dominant topics
- Suggest missing areas: “You have not added anything on Bloom’s Taxonomy”

5. Project Sharing & Import (Phase 2)

- Share project link with peer/tutor
- Import mentor-created test sets

6. Backend Structuring

- Store data with fields: project_id, user_id, unit, tags, content_type, source, created_at

---

Module 4: Discover (Trending, Social Learning Layer)

🔧 Purpose:

To build a socially intelligent layer that shows students what others are asking. This creates curiosity, reduces isolation, and drives deeper exploration through trends.

📈 Core Components:

1. Query Aggregation Engine

- Store all queries across users
- Add subject/Topic tags, query type, timestamp

2. Trending Feed Generator

- Count query frequency per day
- Group using fuzzy match
- Daily top 10 + fallback for low traffic

3. Live Feed (Optional)

- Redis or Firebase cache of last 10 public queries
- Expire after 30 mins

4. Subject Personalization

- Filter feed based on user's exam/subject

5. Follow-up Suggestions

- Based on personal + peer behavior:
    - “Since you asked about X, check this”

6. Sequential Path Mining (Future)

- Track user flow: X → Y → Z topics
- Suggest guided paths or milestone-based plans

7. Feed Privacy Layer

- Anonymized display
- Opt-out toggle per user

Note:
This wireframe has been AI-generated and is intended purely as a **visual reference** to communicate the **initial layout and module structure**.
The final UI will be significantly more **engaging, simplified, and student-friendly**, keeping in mind the needs of Tier 2/3 learners.
We're exploring the use of **Owl doodles, interactive elements, and minimal friction interfaces**.
All modules shown here are part of the **first functional draft** and will be iterated further based on testing, feedback, and design inputs in upcoming phases.