Eyes Unclouded App – Phase 3: Requirements & Dependencies

Author: Khaylub Thompson-Calvin

Updated: April 19, 2025

# Phase 3: Requirements & Dependencies

This phase documents all technical, structural, and software requirements necessary to fully implement the Maximus Perception Tray feature in the Eyes Unclouded App.

## 1. Python Packages & Flask Tools

These should be added to your virtual environment if not already included:  
  
- flask  
- flask-pymongo (or flask-mongoengine if preferred)  
- python-dotenv  
- werkzeug  
- jinja2 (comes with Flask)  
- uuid (to generate unique perception IDs)  
- datetime (for timestamping perception logs)

## 2. MongoDB Requirements

Database: eyes\_unclouded\_db  
Collection: perception\_entries  
  
Fields per entry:  
- \_id: ObjectId  
- user\_id: reference to user  
- chapter\_id: string or ObjectId  
- emotion: string  
- expression\_tag: string (Ekman-based tag)  
- insight: string (user's interpretation)  
- role\_type: string (for feedback logic)  
- timestamp: datetime

## 3. Folder & File Structure Updates

Files to add:  
  
- controllers/book\_controller.py → new Flask blueprint  
- templates/listen.html → audio story player with embedded tray  
- templates/partials/perception\_tray.html → input form  
- views/static/audio/chapters/ → store your chapter .mp3 files  
- views/static/css/perception\_tray.css → optional for styling

## 4. Functional Requirements

- The audio file must successfully load and play via HTML5 <audio> element.  
- After listening or while listening, user can reflect and submit perception.  
- Form submission must validate fields and store entry in MongoDB.  
- Tray UI adapts (optional) to Role Type using CSS or dynamic rendering.  
- After submission, feedback is shown (e.g., quote from Maximus or symbolic animation).

## 5. Recommendations from Project Management Readings

- Define all dependencies in your requirements.txt and keep it versioned in Git.  
- Before adding any new logic, verify your current database is backed up or exported.  
- If multiple features rely on perception data, consider centralizing logic (create perception\_model.py in /models).  
- Track this as a new Sprint Goal: "Integrate Perception Tray module with UI and backend."  
- Use pseudocode before implementation to clarify field flow from form to storage.

## 6. Checklist for This Phase

- [ ] Confirm MongoDB Atlas or Compass DB is connected to app  
- [ ] Add book\_controller.py route blueprint  
- [ ] Add HTML files for listen page and perception tray  
- [ ] Test form UI in browser with test data  
- [ ] Create perception\_entries collection in DB  
- [ ] Add dummy audio chapters to /static/audio/chapters/