Atomicask Technical Documentation

1. Overview / Introduction

Atomicask is a SaaS platform that allows users to **upload documents and interact with them intelligently** through natural-language queries. Once a user uploads a document (such as a PDF, DOCX, or TXT file), Atomicask analyzes its content, extracts the key text data, and enables the user to **ask questions directly about the document** — receiving accurate, context-aware answers in real time.

The system uses advanced text-processing and semantic-search algorithms to understand context, summarize content, and deliver concise answers based on the uploaded material. ccAtomicask helps professionals, students, and organizations save time by **turning static documents into interactive knowledge sources**.

---

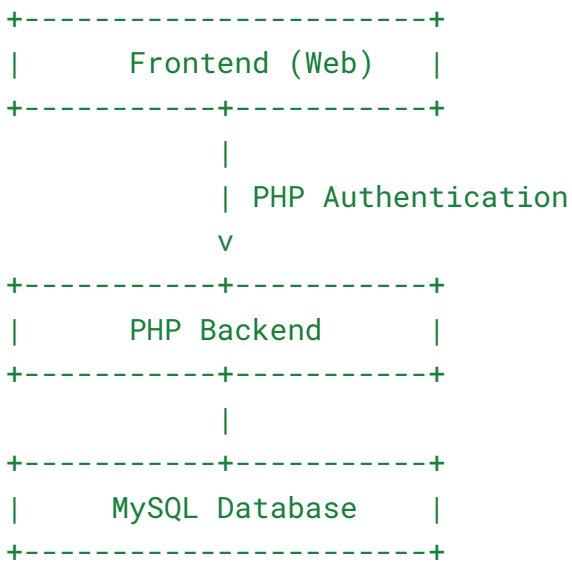
2. System Architecture

Atomicask follows a **modular monolithic architecture** built on PHP and MySQL. The application logic, presentation layer, and API endpoints are handled within a unified backend, while the frontend interfaces (web dashboard, API consumers) communicate through RESTful APIs.

🧩 Components

- **Frontend (Web App)** – Built with HTML, CSS (Tailwind), and JavaScript (Vue or Vanilla JS).
- **Backend (Core API)** – PHP 8.x running on Laravel or a custom MVC framework.
- **Database Layer** – MySQL 8.x for structured data storage.
- **Authentication** – PHP sessions.
- **File Storage** – Local server storage for document uploads.
- **Deployment** – Shared hosting with Nginx/Apache.

🏗️ Architecture Diagram



---

3. Tech Stack

Layer	Technology
Language	PHP 8.x
Database	MySQL 8.x
Frontend	HTML5, CSS3, JavaScript
API Format	REST (JSON)
Deployment	Nginx / Apache
Version Control	Git + GitHub
Storage	Local storage

---

4. Installation and Setup

🧩 Prerequisites

- PHP 8.x or later
- Composer
- MySQL 8.x
- Git

- Node.js (for frontend build if applicable)

⚙️ Installation Steps

1. **Clone the repository**

   ```
   git clone https://github.com/your-org/atomicask.git
   cd atomicask
   ```

2. **Set up your environment**
   - Make sure you have **PHP 8+** and **MySQL** installed.
   - Create a database (e.g., `atomicask_db`) in MySQL.
   - Open the project's configuration file (e.g., `config.php` or any DB config file) and update your credentials:

     ```
     $db_host = "127.0.0.1";

     $db_name = "atomicask_db";

     $db_user = "root";

     $db_pass = "";
     ```

3. **Import the database**
   - Locate the SQL file (e.g., `database.sql`) inside the project.
   - Import it into your database using phpMyAdmin or the command line: `mysql -u root -p atomicask_db < database.sql`
4. **Run the application**

   Start a local PHP server: `php -S localhost:8000`

   - Visit the app in your browser at:
     👉 http://localhost:8000

---

5. API Endpoints

Method	Endpoint	Description	Auth
`POST`	`/api/auth/register`	Register new user	No
`POST`	`/api/auth/login`	Authenticate and get token	No
`GET`	`/api/users/me`	Get current user profile	Yes

GET	/api/tasks	Get all tasks	Yes
POST	/api/tasks	Create new task	Yes
PUT	/api/tasks/{id}	Update task details	Yes
DELETE	/api/tasks/{id}	Delete task	Yes
GET	/api/questions	Fetch list of questions	Yes
POST	/api/questions	Create new question	Yes

Example Response

```
{
  "status": "success",
  "data": {
    "id": 42,
    "title": "Fix email verification bug",
    "assigned_to": "kim@atomicask.com",
    "status": "In Progress"
  }
}
```

---

6. Database Schema

Tables

1. users

Field	Type	Description
id	INT (PK)	Primary key
fullname	VARCHAR(255)	User's full name
email	VARCHAR(255)	Login email
password	VARCHAR(255)	Encrypted password
profilepicture	VARCHAR(255)	Path to profile image

role	ENUM('ADMIN', 'USER')	User role

---

2. chatbot_history

Field	Type	Description
id	INT (PK)	Primary key
chat	TEXT	Chat message content
chatfrom	ENUM('User', 'Assistant')	Message origin
chatownerid	INT (FK → users.id)	Owner of the conversation

---

3. chat_history

Field	Type	Description
id	INT (PK)	Primary key
document_id	INT (FK → documents.id)	Related document
chat	TEXT	Chat message content
chatfrom	ENUM('User', 'Assistant')	Sender type
chatownerid	INT (FK → users.id)	Owner of the chat

---

4. documents

Field	Type	Description
id	INT (PK)	Primary key
name	VARCHAR(255)	Uploaded document name
folder_id	INT (FK → folders.id)	Folder association

`user_id`	INT (FK → users.id)	Owner of the file
`filetext`	LONGTEXT	Extracted or parsed text content
`fileactual`	VARCHAR(255)	File path (e.g., `uploads/...`)

---

5. folders

Field	Type	Description
`id`	INT (PK)	Primary key
`user_id`	INT (FK → users.id)	Folder owner
`name`	VARCHAR(255)	Folder name (e.g “Advertising”)

---

6. logs

Field	Type	Description
`id`	INT (PK)	Primary key
`user_id`	INT (FK → users.id)	User performing the task
`date_created`	DATE	Log creation date
`model_used`	VARCHAR(255)	Model name (e.g., `gpt-4o`, `claude-3-opus`)
`token_used`	INT	Token usage for the session
`texts`	LONGTEXT	Request or result text snippet

---

7. models

Field	Type	Description
`id`	INT (PK)	Primary key

`model`	VARCHAR(255)	Model identifier (e.g., `claude-3-opus-20240229`)

---

8. settings

Field	Type	Description
`id`	INT (PK)	Primary key
`streamtemperature`	FLOAT	Model temperature setting
`chunksize`	INT	Document chunk size
`chunkoverlap`	INT	Overlap between chunks
`systemprompt`	LONGTEXT	System-level prompt or instruction
`userprompt`	LONGTEXT	User-defined custom prompt
`streamingmodel`	VARCHAR(255)	Selected AI model (e.g., `gpt-4o`)
`environment`	VARCHAR(255)	Preset name (e.g., “Highly Detailed”)

✅ **Database Name:** `admin_atomicask`
✅ **Tables:** `users`, `chatbot_history`, `chat_history`, `documents`, `folders`, `logs`, `models`, `settings`
✅ **Total Tables:** 8
✅ **Character Set:** `utf8mb4_general_ci`

---

7. Deployment Guide

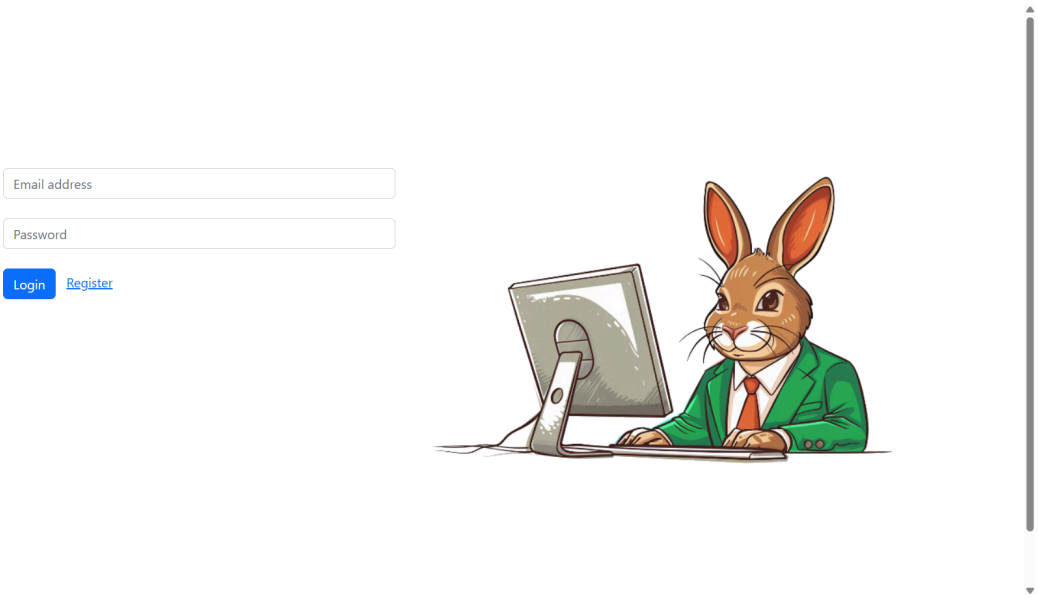
🌐 Manual Server Deployment

1. Upload files to Plesk - Once logged in: Go to Domains -> <domain_name> -> Files
2. Set permission to sensitive files like .env `chmod -R 775 <file>`
3. Install the database to Domains -> <domain_name> -> Databases -> PHPMyAdmin

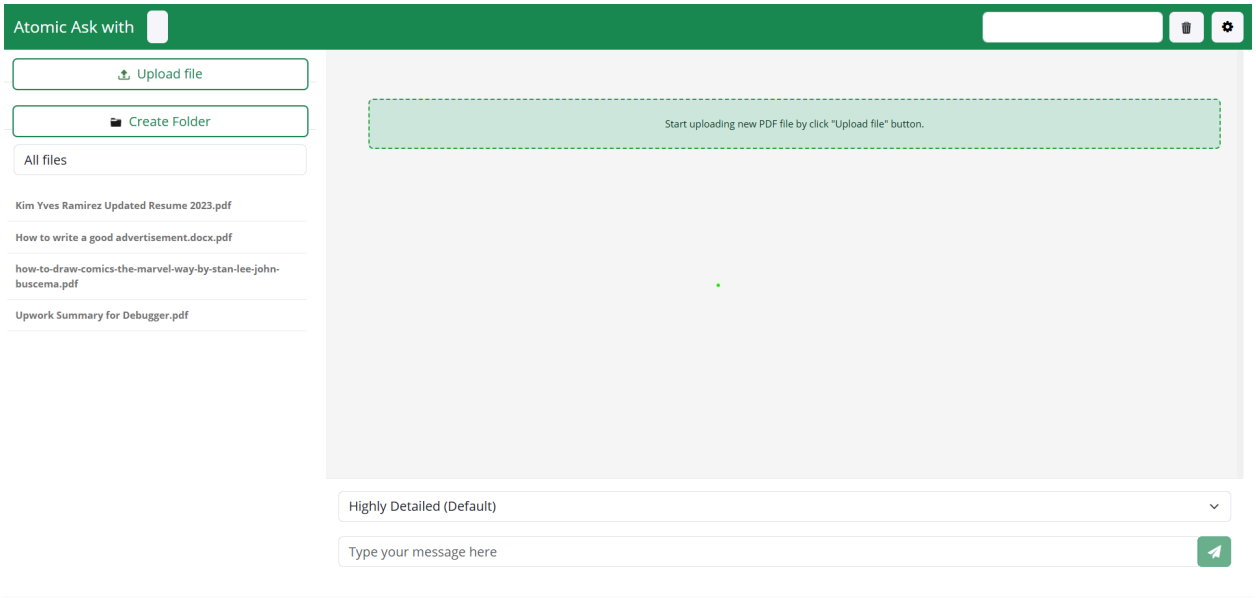
---

8. User Manual

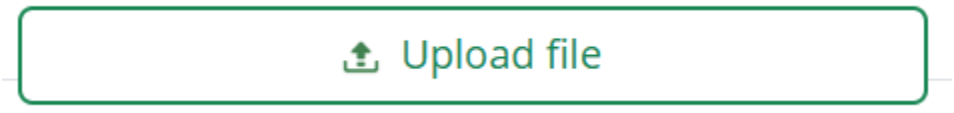
1. Login to https://atomicask.com/



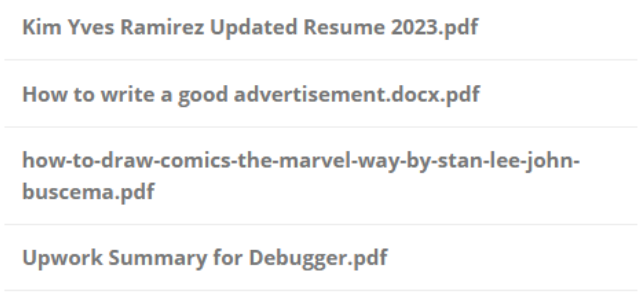
2. This must be the dashboard.



3. To start a new chat, click the Upload file button.

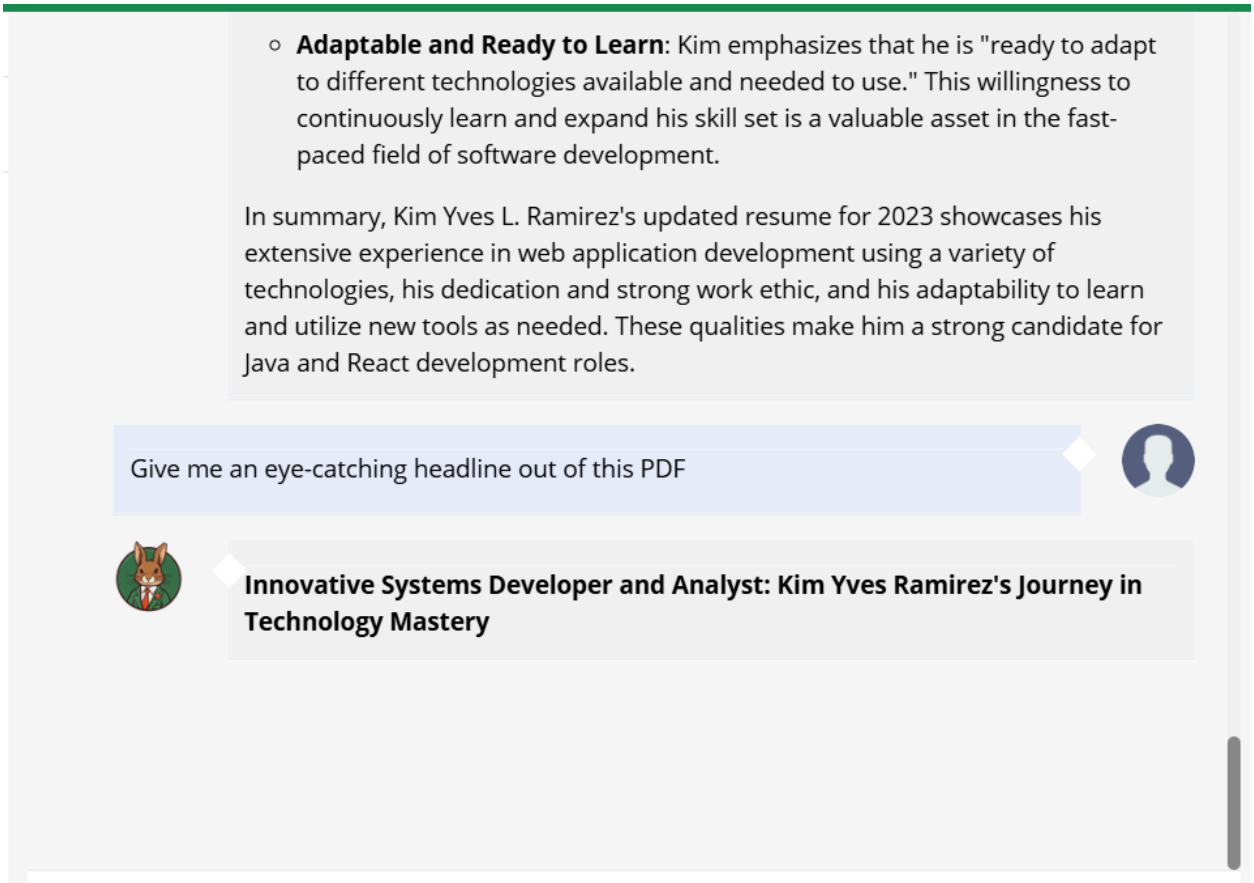


4. To continue with an existing chat, click any of the chat list history on the left panel.



5. The environment selection lets you choose the model and environment that you need for the prompt.

6. After sending the prompt, the center chat panel will be the preview of all of the results of the prompts you sent.



7. This is the message box panel where you can write your prompt related to the document.

8. This is the document preview of the selected chat.

9. This is the delete button where you can delete a chat history

10. This is the settings button where you can adjust and create a new environment.