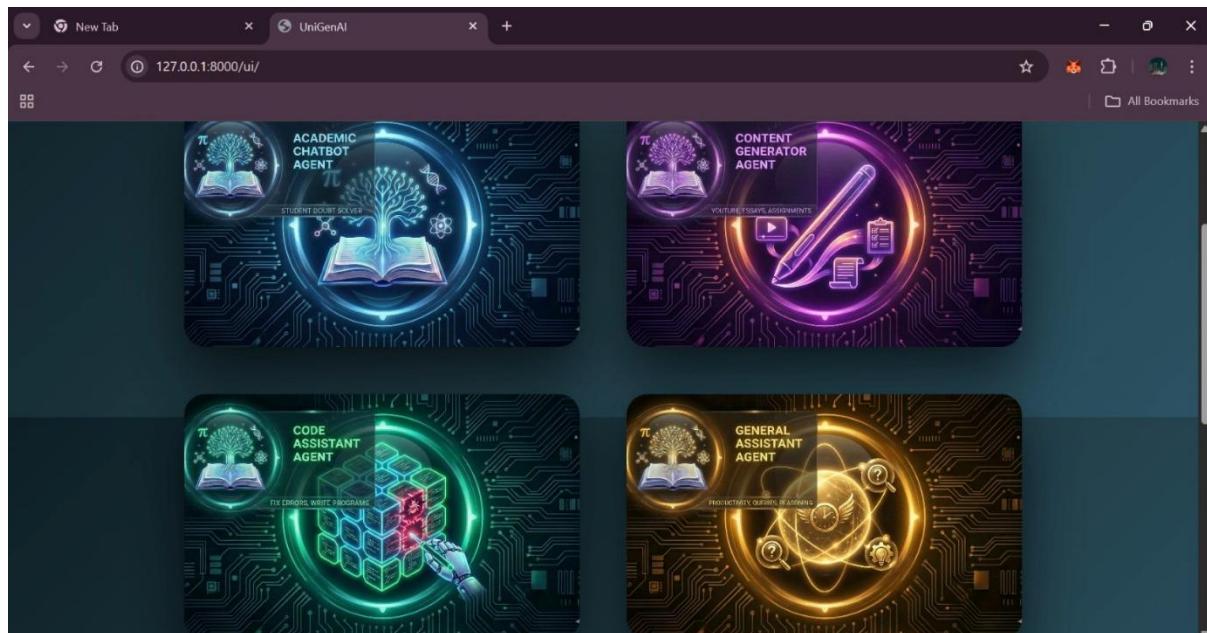
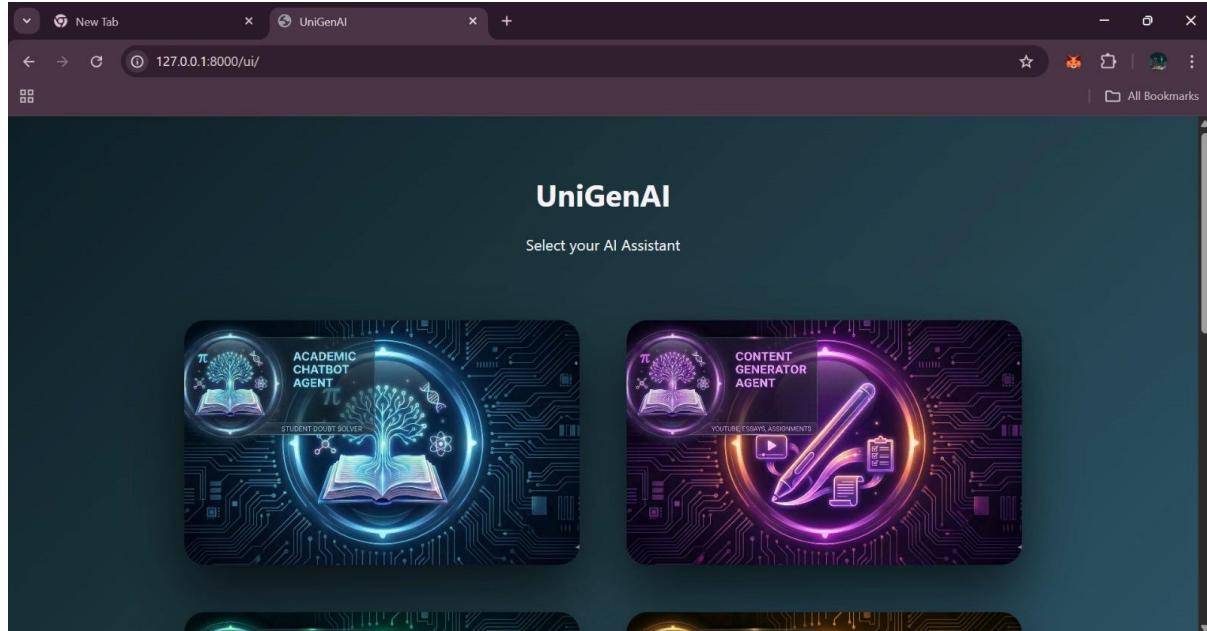


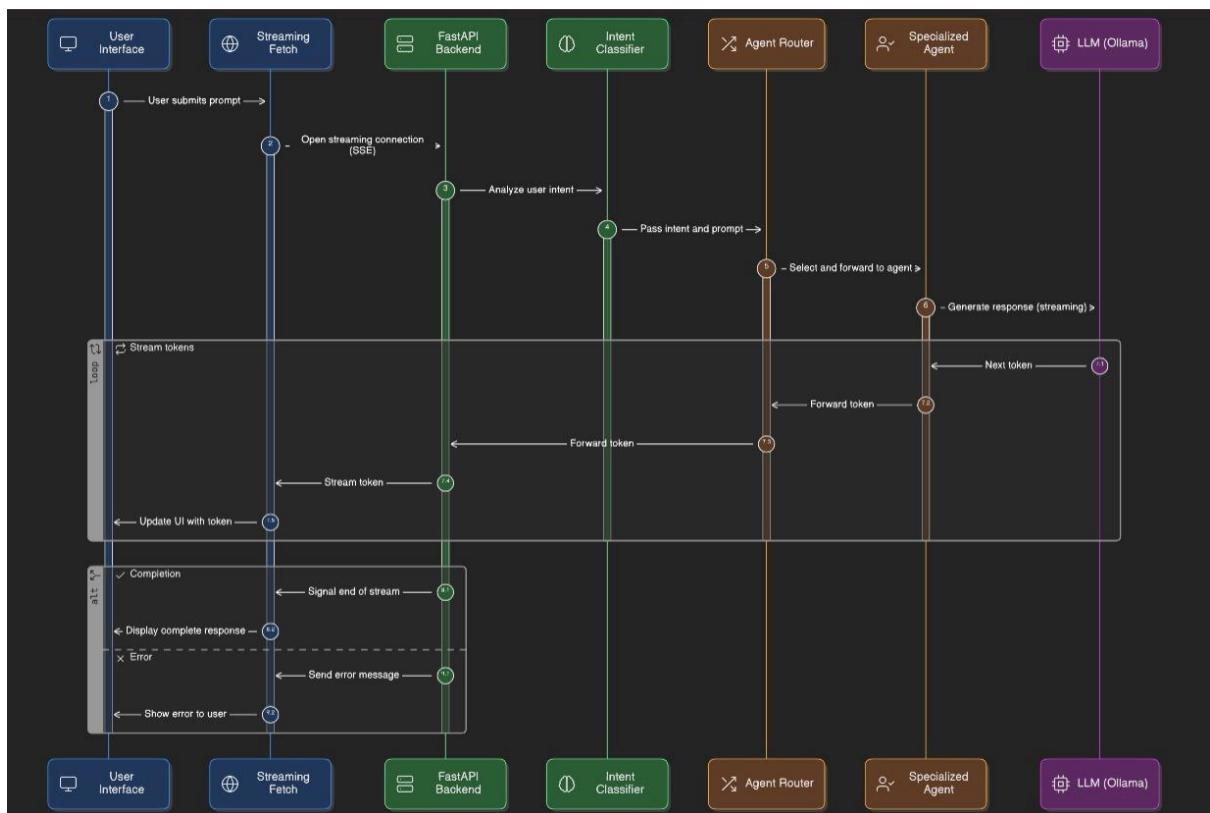
UniGenAI

An Intelligent Multi-Role LLM System for Learning, Content Generation and Software Support

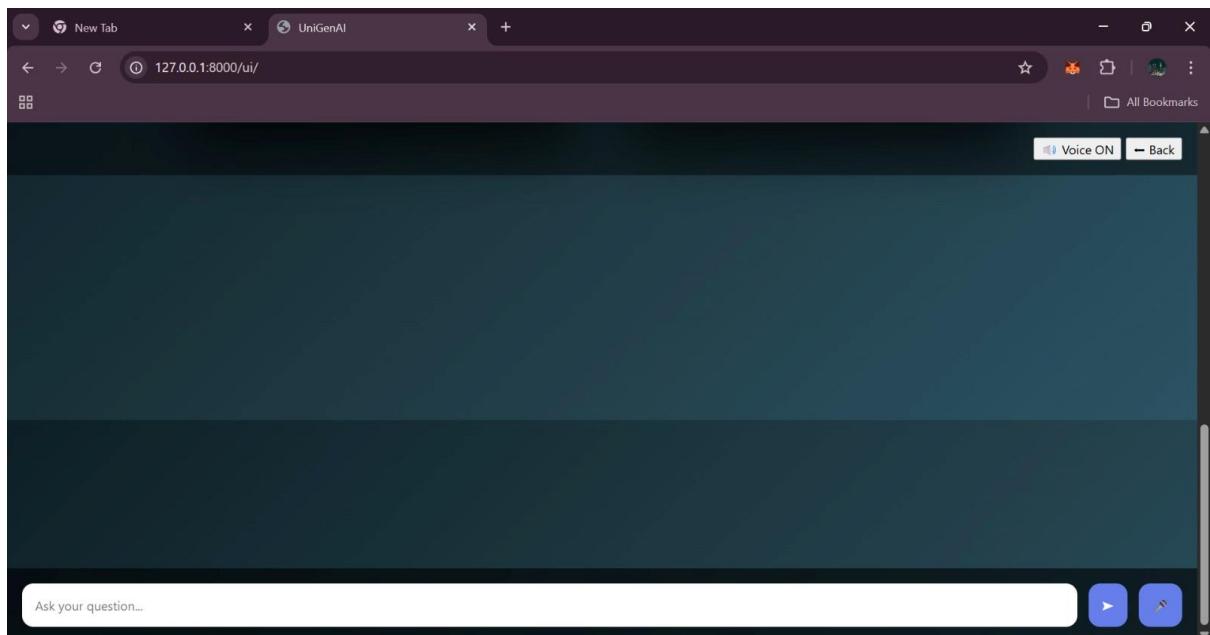
An AI-powered platform for learning, content creation, coding assistance, and interview preparation with advanced user management, real-time streaming, and comprehensive analytics.



Work Flow:

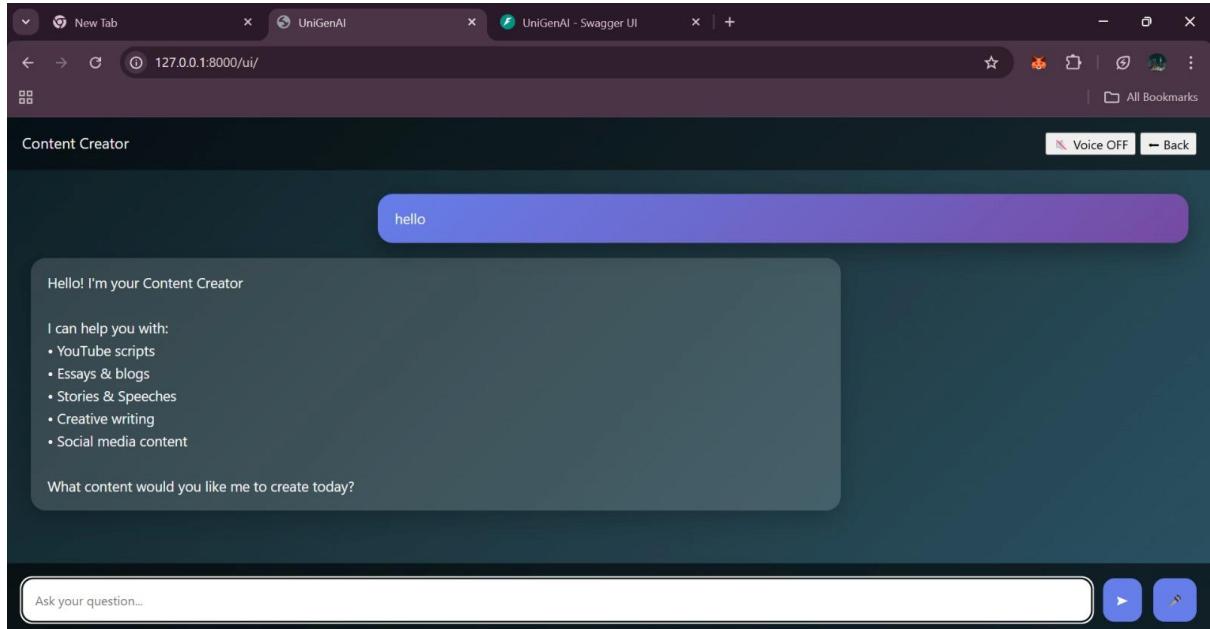


Chatbot Interface

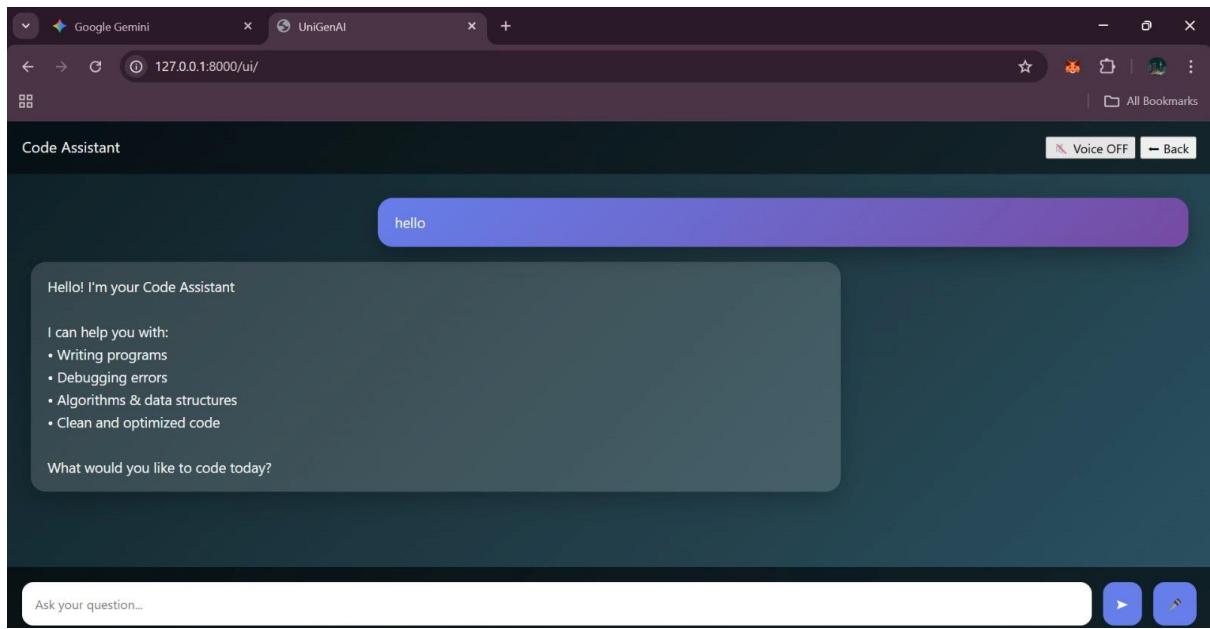


Specialized agents with domain-specific prompts

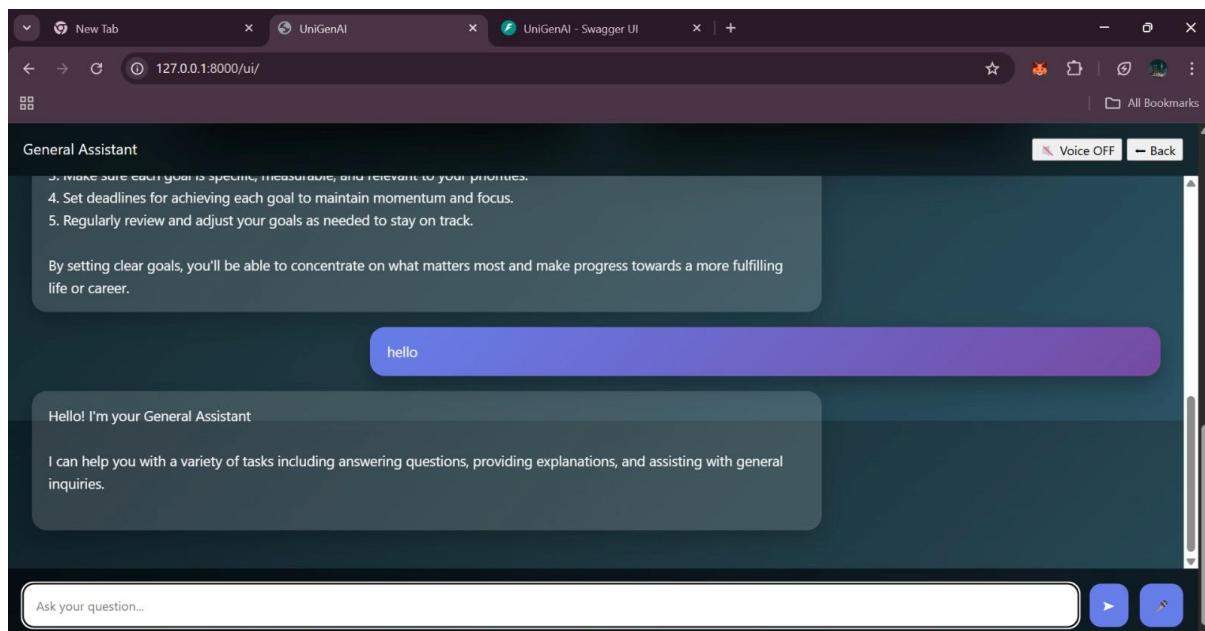
Content (Scripts, Blogs, Stories, Creative writing)



Code (Code explanations, Debugging, Logic walkthrough, Feedback recognition)

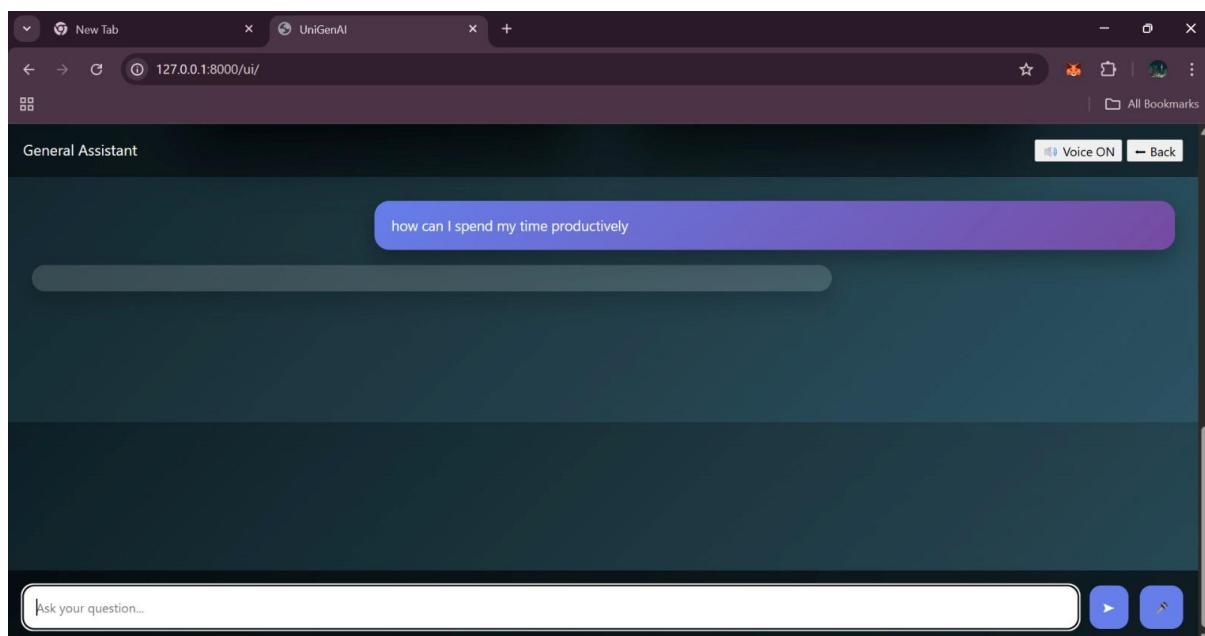


General (Greetings, Casual conversation, Feedback responses)



Intent-aware agent routing

Auto Switch to General Assistant



The screenshot shows a web browser window titled "UniGenAI" at the URL "127.0.0.1:8000/ui". The interface is a dark-themed chat application. At the top, there are tabs for "New Tab" and "UniGenAI", along with standard browser controls like back, forward, and search. A toolbar on the right includes icons for star, refresh, and bookmarks, with a "Voice ON" button currently active.

The main area is labeled "General Assistant". A purple input bar contains the question "how can I spend my time productively". Below it, a text box displays a response: "There are several ways to spend your time productively:" followed by a numbered list of five tips. After the list, another question is asked: "Which area would you like to focus on?". At the bottom, there's a text input field with placeholder text "Ask your question..." and three blue circular buttons with icons for microphone, play, and stop.

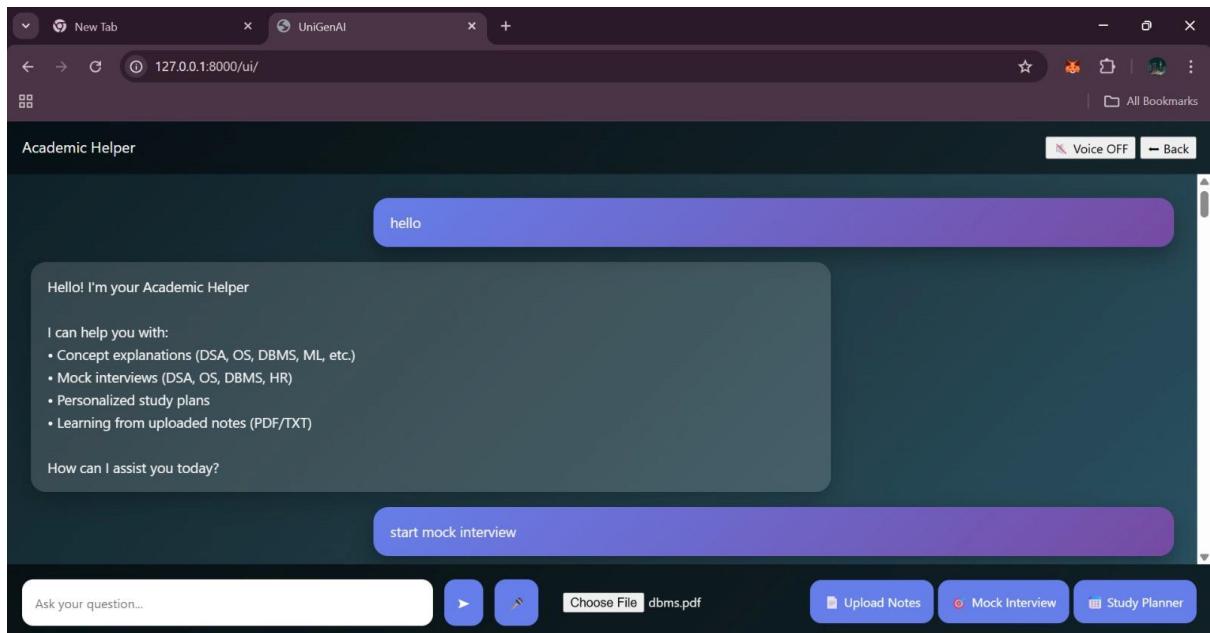
Extend the Conversation

This screenshot continues the conversation from the previous one. The browser title is still "UniGenAI" at "127.0.0.1:8000/ui". The "Voice ON" button has been turned off, indicated by a crossed-out icon.

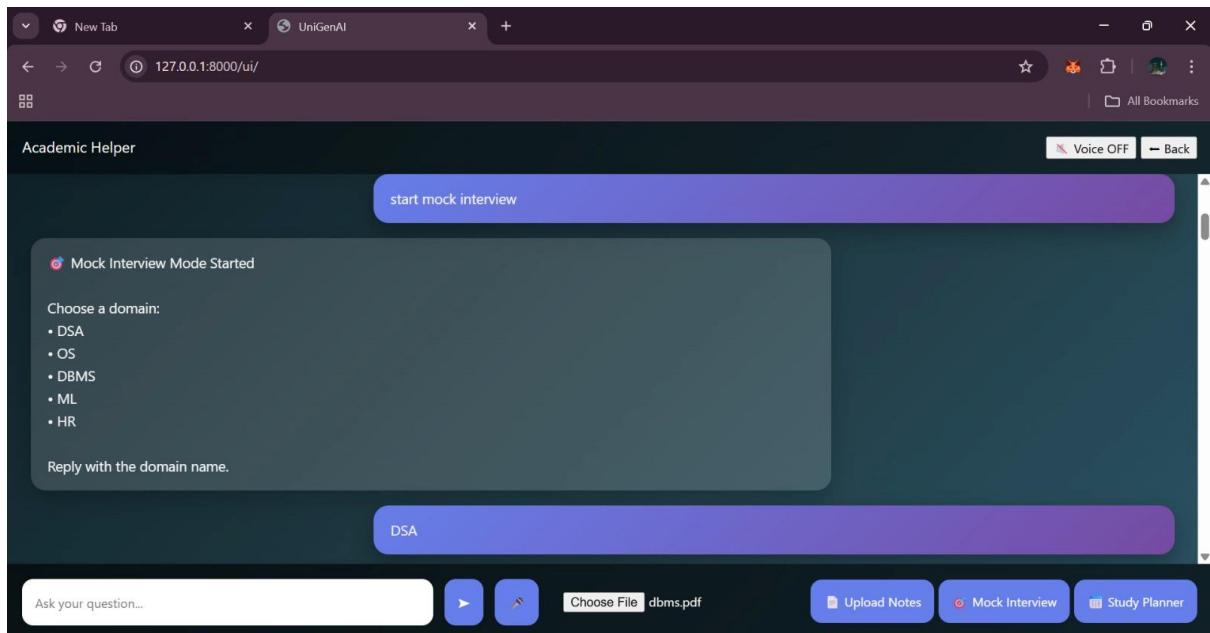
The "General Assistant" section shows the previous question "Which area would you like to focus on?" followed by the AI's response: "1. **Set clear goals**: Identify what you want to achieve and prioritize your tasks accordingly." This response is highlighted with a purple background.

Below this, a new section titled "To set clear goals, consider the following steps:" lists five steps for goal setting. At the bottom of this section, a note states: "By setting clear goals, you'll be able to concentrate on what matters most and make progress towards a more fulfilling life or career." The interface includes the same bottom navigation bar with the "Ask your question..." input field and the three blue circular buttons.

Academic (Subject explanations, Study planner, PDF-based answers (RAG), Mock interview mode, Greeting handling)



Mock interview system with evaluation



New Tab UniGenAI +

127.0.0.1:8000/ui/

Voice OFF | Back

Academic Helper

DSA

Interview Question 1:
What is the difference between an array and a linked list?

Arrays store fixed-size, contiguous blocks of elements for fast index-based access, while linked lists use dynamic, scattered memory with nodes (data + pointer) for flexible insertions/deletions but slower sequential access. The key differences lie in memory (contiguous vs. scattered), size (fixed vs. dynamic), and performance (fast access/slow inserts for arrays; slow access/fast inserts for linked lists).

--- FEEDBACK ---
Evaluation

Strengths:

Ask your question... ► Choose File dbms.pdf Upload Notes Mock Interview Study Planner

New Tab UniGenAI +

127.0.0.1:8000/ui/

Voice OFF | Back

Academic Helper

Strengths:
The candidate provided a clear explanation of the differences between arrays and linked lists, including memory, size, and performance characteristics. They effectively contrasted the fixed-size, contiguous nature of arrays with the dynamic, scattered memory of linked lists.

Weaknesses:
While the candidate covered the key points, their answer lacked specific examples or use cases to illustrate these concepts. Additionally, they did not explicitly define what is meant by "fast access" and "slow inserts," which might lead to confusion.

Key Points:
Arrays have fixed-size, contiguous memory for fast index-based access, while linked lists have dynamic, scattered memory with nodes for flexible insertions/deletions but slower sequential access.

Score: 8/10
Justification:
The candidate demonstrated a good understanding of the fundamental differences between arrays and linked lists. However, to score higher, they would need to provide more concrete examples or clarify ambiguous terms to ensure clarity.

Ask your question... ► Choose File dbms.pdf Upload Notes Mock Interview Study Planner

New Tab UniGenAI +

127.0.0.1:8000/ui/

Voice OFF | Back

Academic Helper

Next Question:
Explain time complexity with an example.

Time complexity is a measure of the amount of computer time it takes an algorithm to run as a function of the length of its input (usually denoted as n). It's a way to assess an algorithm's efficiency and scalability, typically expressed using Big-O notation, which focuses on the worst-case scenario and ignores constant factors and lower-order terms to provide a standardized, hardware-independent metric.

--- FEEDBACK ---
Evaluation

Step 1: Core Technical Concepts
The core technical concepts required for this question are:

* Time complexity
* Big-O notation

Ask your question... ► Choose File dbms.pdf Upload Notes Mock Interview Study Planner

Study Planner

The screenshot shows a web-based study planner titled "Academic Helper". At the top, there's a message from the AI: "Next Question: What is a stack and where is it used?". Below it is a blue button labeled "stop interview". The AI then responds: "Mock interview stopped. How else can I assist you?". Another message follows: "Create a study plan for my interview on 2026-01-24. I can study 6 hours per day." A sidebar on the left is titled "Personalized Study Plan" and shows "Days Remaining: 6". At the bottom, there's a text input field "Ask your question...", a "Choose File" button with "dbms.pdf", and three blue buttons: "Upload Notes", "Mock Interview", and "Study Planner".

This screenshot shows the same "Academic Helper" interface. The AI has generated a detailed study plan: "Create a study plan for my interview on 2026-01-24. I can study 6 hours per day." The "Personalized Study Plan" sidebar now includes "Days Remaining: 6" and "Daily Time Allocation: {'DSA': 2.7, 'OS': 1.6, 'DBMS': 1.7} hours". Below this, under "Study Schedule", there are five sections for "Day 1" through "Day 5", each listing specific study topics and allocated time. The bottom controls remain the same: "Ask your question...", "Choose File" (with "dbms.pdf"), and "Upload Notes", "Mock Interview", "Study Planner" buttons.

This final screenshot shows the most detailed study plan. It includes all the elements from the previous screens, plus additional sections for "Day 2", "Day 3", "Day 4", and "Day 5", each providing a breakdown of the daily study schedule. The bottom of the screen features the familiar "Ask your question...", "Choose File" (with "dbms.pdf"), and "Upload Notes", "Mock Interview", "Study Planner" buttons.

The screenshot shows a dark-themed web browser window for 'UniGenAI' at '127.0.0.1:8000/ui/'. The main content area is titled 'Academic Helper'. It displays a study plan for OS and DBMS, followed by a section for Day 6 (DSA), and concludes with general study tips. At the bottom, there's a text input field 'Ask your question...', a toolbar with icons for back, forward, search, and file operations, and buttons for 'Choose File dbms.pdf', 'Upload Notes', 'Mock Interview', and 'Study Planner'.

- OS (1.6 hrs): Memory Management + concept notes
- DBMS (1.7 hrs): Normalization + SQL practice

Day 6:
- DSA (2.7 hrs): Binary Search Trees + coding practice
- OS (1.6 hrs): Virtual Memory + concept notes
- DBMS (1.7 hrs): Transactions & ACID + SQL practice

To create an effective study plan for your interview on 2026-01-24, I recommend the following:

1. **Duration:** Allocate 6 hours per day for studying.
2. **Frequency:** Study every day from now until the interview date (approximately 3 months).
3. **Content Focus:**
 - * Review relevant course materials and notes (30% of study time)
 - * Practice past questions or exam simulations (40% of study time)
 - * Focus on weak areas or topics that require improvement (20% of study time)
 - * Take practice tests or quizzes to assess knowledge and identify areas for improvement (10% of study time)

Ask your question... Choose File dbms.pdf Upload Notes Mock Interview Study Planner

Retrieval-Augmented Generation (RAG)

This screenshot shows the UniGenAI interface after a PDF named 'dbms.pdf' has been uploaded. The AI has generated a simplified explanation of DBMS based on the uploaded content. The interface includes a toolbar with various icons and buttons for interacting with the AI and managing files.

from the pdf uploaded explain the dbms in simple terms

Based on the provided PDF, here's a simplified explanation of DBMS:

What is DBMS?

dbms.pdf uploaded successfully

Ask your question... Choose File dbms.pdf Upload Notes Mock Interview Study Planner

This screenshot shows the UniGenAI interface displaying a detailed breakdown of DBMS concepts, generated from the uploaded PDF. It includes sections on what DBMS is, key concepts, and specific definitions for Database, Table, Record, and Field.

from the pdf uploaded explain the dbms in simple terms

Based on the provided PDF, here's a simplified explanation of DBMS:

What is DBMS?

DBMS stands for Database Management System. It's a software system that helps you manage and organize data in a database.

Key Concepts:

1. **Database**: A collection of organized data.
2. **Table**: A table is a way to store data in a database. Each row represents a single record, and each column represents a field or attribute of that record.
3. **Record**: A record is a single row in a table.
4. **Field**: A field is a single column in a table.

Ask your question... Choose File dbms.pdf Upload Notes Mock Interview Study Planner

The screenshot shows a web browser window with the URL 127.0.0.1:8000/ui/. The page title is "Academic Helper". The main content area displays information under the heading "DBMS Functions". It includes sections for "Data Storage", "Data Retrieval", "Data Manipulation", and "Security". Below this, there are sections for "DBMS Components" (System Catalog, Metadata, Data Warehouse) and "DBMS Operations" (Selection, Projection, Intersection, Cartesian Product). A note at the bottom states: "In summary, a DBMS is a software system that helps you manage, organize, and retrieve data in a database, providing benefits such as efficient storage and retrieval, improved security, and scalability." At the bottom of the page are buttons for "Ask your question...", "Choose File" (with "dbms.pdf" selected), "Upload Notes", "Mock Interview", and "Study Planner".

This screenshot shows the same web browser window with the URL 127.0.0.1:8000/ui/. The page title is "Academic Helper". The main content area now displays information under the heading "Benefits". It includes sections for "Data Organization", "Efficient Storage and Retrieval", "Improved Security", and "Scalability". A note at the bottom states: "In summary, a DBMS is a software system that helps you manage, organize, and retrieve data in a database, providing benefits such as efficient storage and retrieval, improved security, and scalability." At the bottom of the page are buttons for "Ask your question...", "Choose File" (with "dbms.pdf" selected), "Upload Notes", "Mock Interview", and "Study Planner".

Automatic agent switching without losing UI context

Auto switch to content creator

This screenshot shows the same web browser window with the URL 127.0.0.1:8000/ui/. The page title is "Content Creator". The main content area displays information under the heading "Content Creator". It includes sections for "Benefits" (which lists the same four points as the previous screenshots) and a note at the bottom stating: "In summary, a DBMS is a software system that helps you manage, organize, and retrieve data in a database, providing benefits such as efficient storage and retrieval, improved security, and scalability." At the bottom of the page is a large input field containing the text "for my youtube video on AI Fundamentals provide me a script". At the very bottom are buttons for "Ask your question...", "Choose File", "Upload Notes", "Mock Interview", and "Study Planner".

New Tab UniGenAI +

127.0.0.1:8000/ui/

Content Creator

Voice ON Back

for my you tube video on AI Fundamentals provide me a script

Here's a script for your YouTube video on "AI Fundamentals":

Title: "Unlocking the Power of Artificial Intelligence: A Beginner's Guide to AI Fundamentals"

Intro (0:00 - 0:30)

(Upbeat background music starts playing. You appear on screen with a friendly smile)

You: "Hey everyone, welcome back to [Your Channel Name]! Are you curious about Artificial Intelligence? Do you want to learn how AI can transform your life and business? Well, you're in the right place! In this video, we'll dive into the world of AI fundamentals. By the end of this tutorial, you'll have a solid understanding of what AI is, its types, and how it works. So let's get started!"

Section 1: What is Artificial Intelligence? (0:30 - 2:00)

Ask your question...  

New Tab UniGenAI +

127.0.0.1:8000/ui/

Content Creator

Voice OFF Back

(Visuals of AI-related images appear on screen)

You: "So, what exactly is Artificial Intelligence? Simply put, AI refers to the development of computer systems that can perform tasks that would typically require human intelligence. These tasks include learning, problem-solving, decision-making, and more. But how does it do all this?"

(Visuals of a brain or a robot appear on screen)

You: "Well, AI is based on algorithms and data, which allows computers to make decisions and take actions without being explicitly programmed. It's like having a super-smart computer that can learn from its mistakes and adapt to new situations."

Section 2: Types of AI (2:00 - 4:30)

(Visuals of different types of AI appear on screen)

You: "So, there are several types of AI out there. Let's talk about the main ones:

Ask your question...  

New Tab UniGenAI +

127.0.0.1:8000/ui/

Content Creator

Voice OFF Back

1. **Narrow or Weak AI**: This type of AI is designed to perform a specific task, like facial recognition or image analysis.
2. **General or Strong AI**: This type of AI has the ability to perform any intellectual task that humans can, which is still a topic of debate among experts.
3. **Superintelligence**: This type of AI surpasses human intelligence in all domains and has the potential to revolutionize our world.

(Visuals of each type of AI appear on screen)

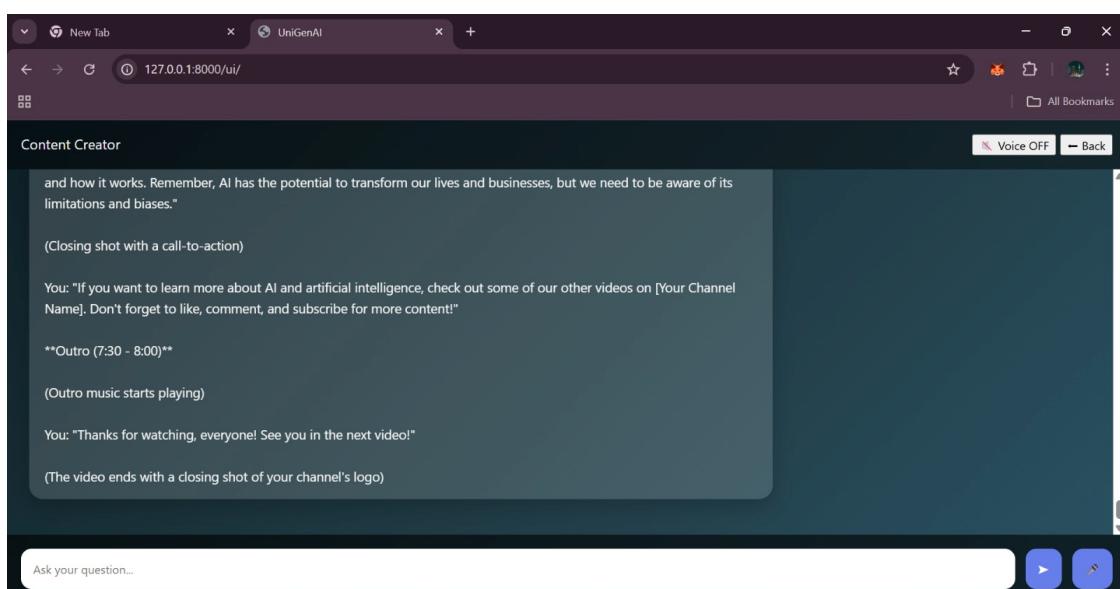
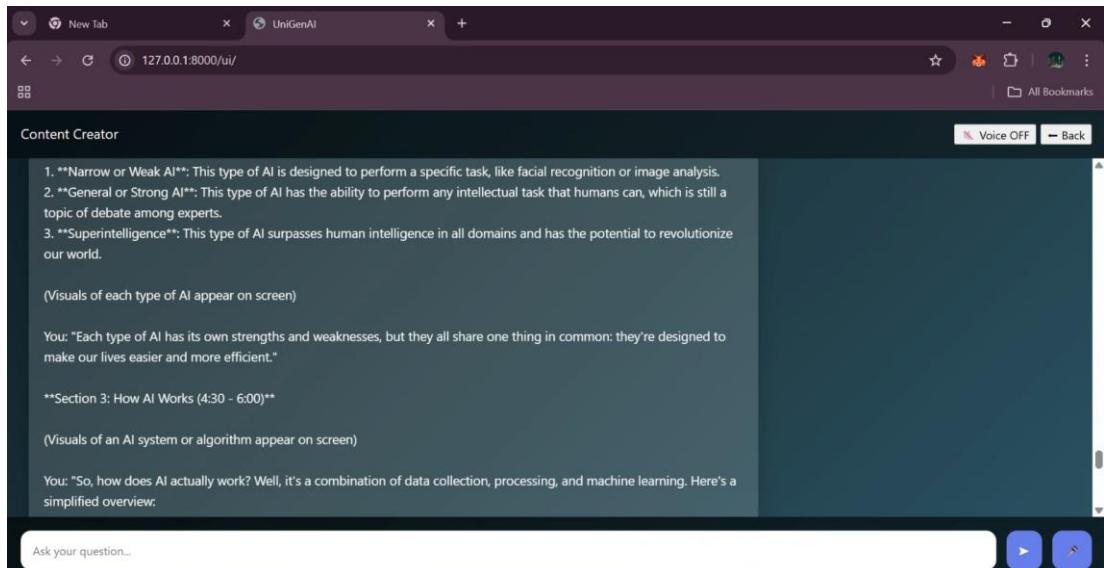
You: "Each type of AI has its own strengths and weaknesses, but they all share one thing in common: they're designed to make our lives easier and more efficient."

Section 3: How AI Works (4:30 - 6:00)

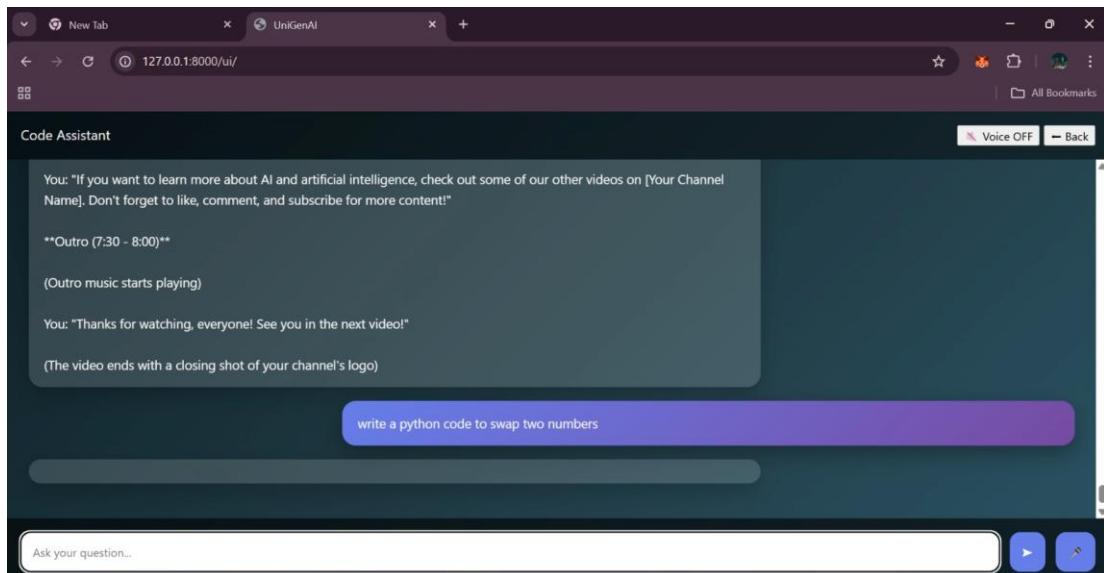
(Visuals of an AI system or algorithm appear on screen)

You: "So, how does AI actually work? Well, it's a combination of data collection, processing, and machine learning. Here's a simplified overview:

Ask your question...  



Auto Switch to Code Assistant



The goal is to write a function that swaps two numbers without using a temporary variable. Here's the step-by-step logic:

1. Take two numbers as input.
2. Use arithmetic operations to achieve swapping without a temp variable.

Code Solution

```
```python
def swap_numbers(a, b):
 # Using arithmetic operations to swap
 a = a + b
 b = a - b
 return a, b
```
Ask your question...
```

```
a = a - b
return a, b

# Example usage:
num1 = 10
num2 = 20

print("Before swapping: num1 =", num1, "num2 =", num2)

num1, num2 = swap_numbers(num1, num2)

print("After swapping: num1 =", num1, "num2 =", num2)
```
Ask your question...
```

In the original solution, we had issues with variable assignment. The initial intention was to use arithmetic operations to achieve swapping without a temporary variable. However, there were errors in the implementation.

- In the line `a = a + b`, it should have been `temp = a` before `a = a + b`. This is because we're trying to swap the values but end up with an incorrect result due to losing the original value of 'a'.
- The same issue was present in `b = a - b`, where again, `temp = a` would be required.

The corrected version above fixes this by maintaining both variables' original values while updating them accordingly using arithmetic operations.

```
print("After swapping: num1 =", num1, "num2 =", num2)
```
Ask your question...
```

Auto Switch to Academic Helper

In the original solution, we had issues with variable assignment. The initial intention was to use arithmetic operations to achieve swapping without a temporary variable. However, there were errors in the implementation.

- In the line `a = a + b`, it should have been `temp = a` before `a = a + b`. This is because we're trying to swap the values but end up with an incorrect result due to losing the original value of `a`.
- The same issue was present in `b = a - b`, where again, `temp = a` would be required.

The corrected version above fixes this by maintaining both variables' original values while updating them accordingly using arithmetic operations.

As a engineer student how can I spend my day productively

Ask your question... Choose File dbms.pdf Upload Notes Mock Interview Study Planner

As an engineering student, here's a suggested plan to help you spend your day productively:

****Morning Routine (8:00 AM - 9:00 AM)****

1. **Wake up early**: Get a head start on the day by waking up at least an hour before the rest of your family.
2. **Exercise or stretch**: Engage in some light physical activity, such as yoga or a short walk, to get your blood flowing and energy levels up.
3. **Shower and dress**: Freshen up and get dressed for the day.

****Study and Learning (9:00 AM - 12:00 PM)****

1. **Review notes and textbooks**: Go through your class notes, textbook, and any other study materials to review key concepts and topics.
2. **Focus on challenging subjects**: Identify areas where you need improvement and focus on those subjects.

As a engineer student how can I spend my day productively

Ask your question... Choose File dbms.pdf Upload Notes Mock Interview Study Planner

3. **Practice problems and examples**: Work on practice problems, examples, or past exams to reinforce your understanding.

****Lunch Break (12:00 PM - 1:00 PM)****

1. **Take a break**: Give yourself time to relax and recharge.
2. **Eat a nutritious meal**: Fuel your body with a healthy lunch that includes plenty of fruits, vegetables, whole grains, lean proteins, and healthy fats.

****Group Study or Collaboration (1:00 PM - 3:00 PM)****

1. **Meet up with study group**: Gather with classmates or peers to discuss challenging topics, share notes, and collaborate on projects.
2. **Work on group projects**: Make progress on group assignments or projects that require multiple perspectives and skills.

****Self-Development and Relaxation (3:00 PM - 5:00 PM)****

Voice OFF Back

Ask your question... Choose File dbms.pdf Upload Notes Mock Interview Study Planner

The screenshot shows a dark-themed web application window titled "Academic Helper". The main content area displays a list of tasks:

1. **Learn a new skill or hobby**: Spend time learning a new programming language, musical instrument, or other activity that interests you.
2. **Read industry publications or blogs**: Stay up-to-date with the latest developments in your field by reading industry publications, blogs, or news sites.
3. **Take a short walk or stretch**: Get some fresh air and take a break to stretch your legs.

Below this is a section titled "Evening Routine (5:00 PM - 9:00 PM)":

1. **Review what you've learned**: Go over the day's material, highlighting important concepts and key takeaways.
2. **Work on assignments or projects**: Make progress on any outstanding assignments or projects that require your attention.
3. **Wind down and relax**: Engage in a relaxing activity, such as meditation, listening to music, or watching a funny video.

At the bottom, there is an "Additional Tips" section with the following bullet points:

- * Set specific goals for the day and prioritize them.
- * Use a planner, calendar, or app to stay organized and on track.

At the very bottom of the content area, a message reads: "By following this schedule, you'll be able to make the most of your day as an engineering student. Remember to stay flexible and adjust the plan as needed to suit your individual needs and preferences!"

At the bottom of the window, there is a search bar with placeholder text "Ask your question...", a file upload button labeled "Choose File dbms.pdf", and three buttons for "Upload Notes", "Mock Interview", and "Study Planner".

This screenshot is nearly identical to the one above, showing the same "Academic Helper" content and layout. The main difference is the addition of a large, semi-transparent callout box at the bottom of the content area. This box contains the following text:

By following this schedule, you'll be able to make the most of your day as an engineering student. Remember to stay flexible and adjust the plan as needed to suit your individual needs and preferences!

The rest of the interface, including the search bar, file upload, and buttons, remains the same.

Swagger UI

The screenshot shows a browser window titled "UniGenAI - Swagger UI" with the URL "127.0.0.1:8000/docs". The page displays the UniGenAI API documentation. At the top, it shows the version "0.1.0" and "OAS 3.1". Below this, there is a brief description: "Multi-Agent LLM System (Academic | Code | Content | General)".

The main content area is titled "default" and lists several API endpoints:

- POST /upload-pdf** Upload Pdf
- POST /chat** Chat
- POST /api/user/create** Create User Endpoint
- GET /api/user/{user_id}** Get User
- GET /api/users/all** Get All Users
- POST /api/interview/save** Save Interview Result

The screenshot shows a browser window with the address bar set to 127.0.0.1:8000/docs. The main content area displays a list of API endpoints under the heading "UniGenAI - Swagger UI". The endpoints are categorized by color-coded boxes:

- Green box: GET /api/interview/history/{user_id} Get User Interview History
- Green box: GET /api/interview/stats/{user_id} Get User Stats
- Green box: POST /api/planner/save Save Planner
- Green box: GET /api/planner/{user_id} Get Plans
- Orange box: PUT /api/planner/{plan_id}/update Update Planner
- Light blue box: GET /api/chat/history/{user_id} Get User Chat History
- Light blue box: GET /api/debug/interviews/{user_id} Debug Interviews
- Pink box: DELETE /api/debug/interviews/{user_id} Delete Interviews

Below the list, there is a section titled "Schemas" which is currently empty.

Schemas

The screenshot shows the same browser window with the address bar set to 127.0.0.1:8000/docs. The main content area now displays the "Schemas" section. It lists several schema definitions:

- Body_upload_pdf_upload_pdf_post > Expand all object
- ChatRequest > Expand all object
- HTTPValidationError > Expand all object
- InterviewResultRequest > Expand all object
- StudyPlanRequest > Expand all object
- ValidationError > Expand all object

Real-time streaming using Server-Sent Events (SSE)

POST - Chat

The screenshot shows the browser window with the address bar set to 127.0.0.1:8000/docs#/default/chat_chat_post. The main content area is a configuration form for the "POST /chat" endpoint:

- Parameters**: A table with one row for "user_id":

| Name | Description |
|---------|--------------------|
| user_id | integer
(query) |
- Request body**: Required. Type: application/json. The "Edit Value" field contains the following JSON:

```
{  "message": "Explain PCA in simple terms",  "forced_role": "academic"}}
```

The screenshot shows the UniGenAI - Swagger UI interface. A POST request is being made to the endpoint `/chat`. The request body contains the following JSON:

```
curl -X 'POST' \
  'http://127.0.0.1:8000/chat' \
  -H 'Accept: application/json' \
  -H 'Content-type: application/json' \
-d '{
  "message": "Explain PCA in simple terms",
  "forced_role": "academic"
}'
```

The response code is 200, and the response body is:

```
data: {"token": "", "agent": "academic"}  
data: {"token": "PCA", "agent": "academic"}  
data: {"token": " stands", "agent": "academic"}
```

The screenshot shows the UniGenAI - Swagger UI interface. A POST request is being made to the endpoint `/chat`. The response body is a large list of tokens and their agents:

```
data: {"token": "PCA", "agent": "academic"}  
data: {"token": " stands", "agent": "academic"}  
data: {"token": " for", "agent": "academic"}  
data: {"token": " Principal", "agent": "academic"}  
data: {"token": " Component", "agent": "academic"}  
data: {"token": " Analysis", "agent": "academic"}  
data: {"token": ",", "agent": "academic"}  
data: {"token": " which", "agent": "academic"}  
data: {"token": " is", "agent": "academic"}  
data: {"token": " a", "agent": "academic"}  
data: {"token": " statistical", "agent": "academic"}  
data: {"token": " technique", "agent": "academic"}  
data: {"token": " used", "agent": "academic"}
```

The response headers are:

```
access-control-allow-credentials: true  
access-control-allow-origin: *  
cache-control: no-cache  
connection: keep-alive  
content-type: text/event-stream; charset=utf-8  
date: Sun, 18 Jan 2016 07:39:09 GMT  
server: uvicorn  
transfer-encoding: chunked  
x-accel-buffering: no
```

The screenshot shows the UniGenAI - Swagger UI interface. It displays two examples of responses:

Successful Response

Code: 200
Description: Successful Response
Media type: application/json
Example Value: string

Validation Error

Code: 422
Description: Validation Error
Media type: application/json
Example Value: {
 "detail": [
 {
 "loc": [
 "string",
 @
],
 "msg": "string",
 "type": "string"
 }
]
}

GET (All Users)

The screenshot shows the UniGenAI - Swagger UI interface for the `GET /api/users/all` endpoint. The top navigation bar shows the URL `127.0.0.1:8000/docs#/default/get_all_users_api_users_all_get`. The main content area has a title `GET /api/users/all GetAll Users`. Below it is a description: "Get all users in the system". A "Parameters" section indicates "No parameters". At the bottom are "Execute" and "Clear" buttons. In the "Responses" section, there is a "Curl" code block:

```
curl -X 'GET' \
  'http://127.0.0.1:8000/api/users/all' \
  -H 'accept: application/json'
```

Below the curl block is a "Request URL" field containing `http://127.0.0.1:8000/api/users/all`. The second part of the screenshot shows the response details for a 200 status code. It includes a "Response body" section with a JSON array of user objects and a "Response headers" section with the following content:

```
content-length: 658
content-type: application/json
date: Sun,18 Jan 2026 07:55:20 GMT
server: uvicorn
```

GET (All User ID)

The screenshot shows the UniGenAI - Swagger UI interface for the `GET /api/user/{user_id}` endpoint. The top navigation bar shows the URL `127.0.0.1:8000/docs#/default/get_user_api_user_user_id_get`. The main content area has a title `GET /api/user/{user_id} Get User`. Below it is a description: "Get user information by ID". A "Parameters" section shows a required parameter `user_id` of type integer with a value of 1. At the bottom are "Execute" and "Clear" buttons. In the "Responses" section, there is a "Curl" code block:

```
curl -X 'GET' \
```

Request URL
http://127.0.0.1:8000/api/user/1

Server response

Code Details

200 Response body

```
{ "id": 1, "username": "sonali", "created_at": "2026-01-16T17:29:35.211754" }
```

Response headers

```
content-length: 70  
content-type: application/json  
date: Sun, 18 Jan 2026 07:57:24 GMT  
server: uvicorn
```

Responses

| Code | Description | Links |
|------|---------------------|----------|
| 200 | Successful Response | No links |

Media type

application/json

Controls Accept header.

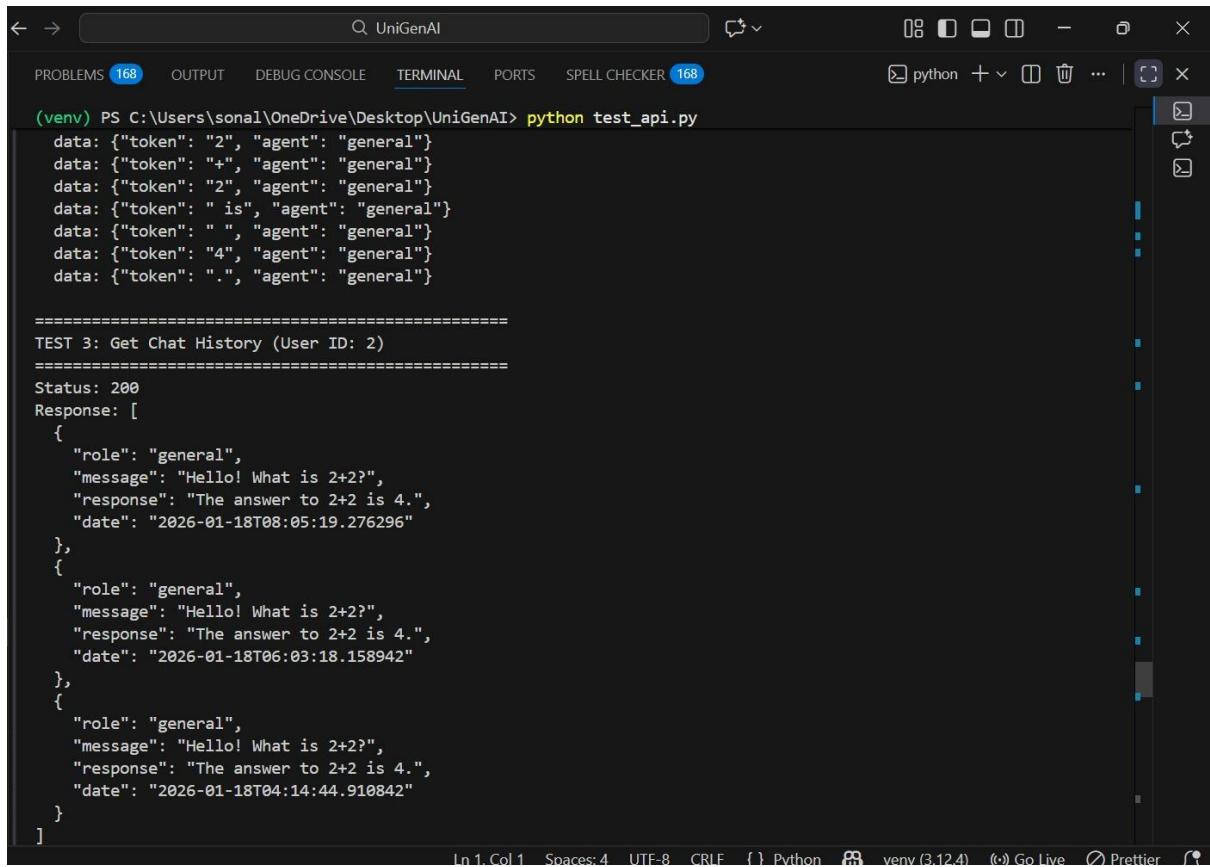
TESTING

API Working

```
(venv) PS C:\Users\sonal\OneDrive\Desktop\UniGenAI> python test_api.py
=====
TEST 1: Create User
=====
Status: 200
Response: {
    "id": 2,
    "username": "test_user_1"
}

=====
TEST 2: Send Chat Message (User ID: 2)
=====
Status: 200
Response (streaming):
  data: {"token": "", "agent": "general"}
  data: {"token": "The", "agent": "general"}
  data: {"token": " answer", "agent": "general"}
  data: {"token": " to", "agent": "general"}
  data: {"token": " ", "agent": "general"}
  data: {"token": "2", "agent": "general"}
  data: {"token": "+", "agent": "general"}
  data: {"token": "2", "agent": "general"}
  data: {"token": " is", "agent": "general"}
  data: {"token": " ", "agent": "general"}
  data: {"token": "4", "agent": "general"}
  data: {"token": ".", "agent": "general"}

=====
TEST 3: Get Chat History (User ID: 2)
=====
Status: 200
Response: [
```



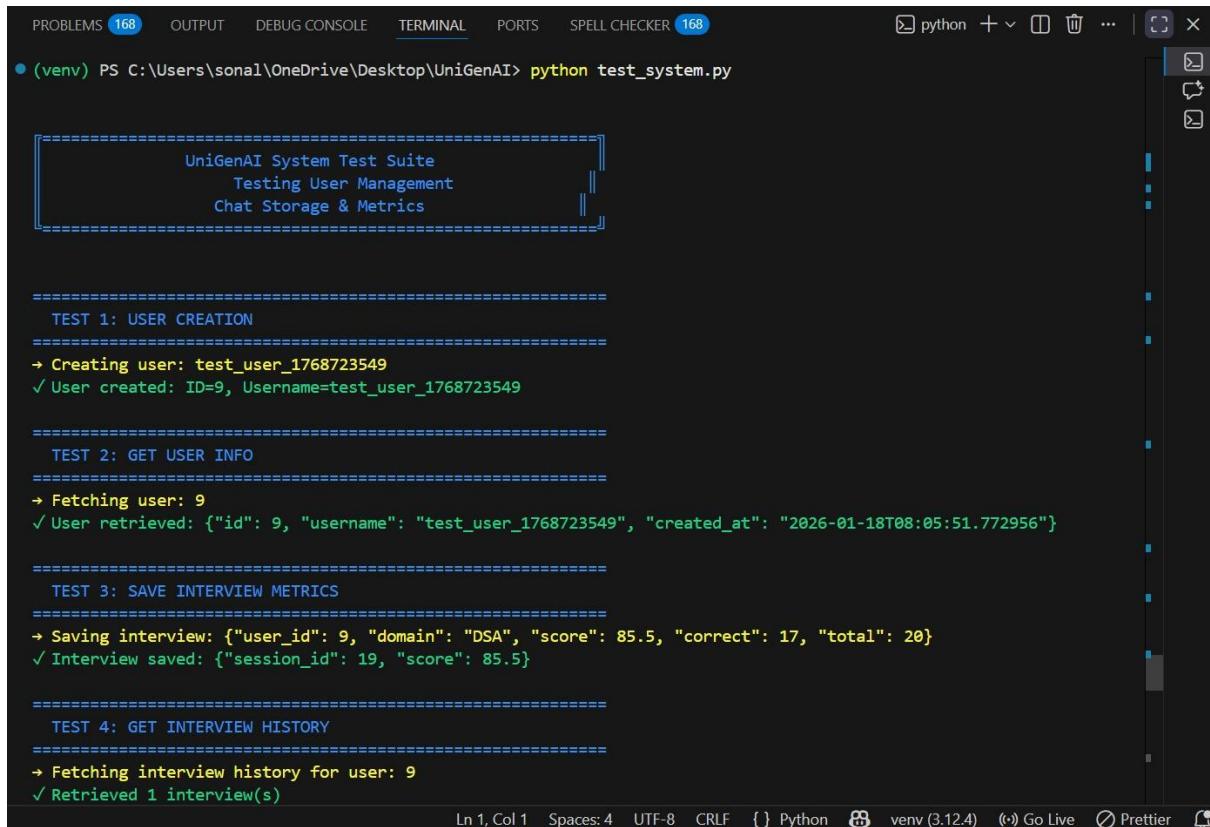
UniGenAI

PROBLEMS 168 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 168

(venv) PS C:\Users\sonal\OneDrive\Desktop\UniGenAI> python test_api.py

```
data: {"token": "2", "agent": "general"}  
data: {"token": "+", "agent": "general"}  
data: {"token": "2", "agent": "general"}  
data: {"token": "is", "agent": "general"}  
data: {"token": " ", "agent": "general"}  
data: {"token": "4", "agent": "general"}  
data: {"token": ".", "agent": "general"}  
  
=====  
TEST 3: Get Chat History (User ID: 2)  
=====  
Status: 200  
Response: [  
  {  
    "role": "general",  
    "message": "Hello! What is 2+2?",  
    "response": "The answer to 2+2 is 4.",  
    "date": "2026-01-18T08:05:19.276296"  
  },  
  {  
    "role": "general",  
    "message": "Hello! What is 2+2?",  
    "response": "The answer to 2+2 is 4.",  
    "date": "2026-01-18T06:03:18.158942"  
  },  
  {  
    "role": "general",  
    "message": "Hello! What is 2+2?",  
    "response": "The answer to 2+2 is 4.",  
    "date": "2026-01-18T04:14:44.910842"  
  }]  
Ln 1 Col 1 Spaces: 4 UTF-8 CRLF { } Python 8 venv (3.12.4) (o) Go Live (o) Prettier
```

System Working



UniGenAI System Test Suite

Testing User Management

Chat Storage & Metrics

=====

TEST 1: USER CREATION

=====

```
> Creating user: test_user_1768723549  
✓ User created: ID=9, Username=test_user_1768723549
```

=====

TEST 2: GET USER INFO

=====

```
> Fetching user: 9  
✓ User retrieved: {"id": 9, "username": "test_user_1768723549", "created_at": "2026-01-18T08:05:51.772956"}
```

=====

TEST 3: SAVE INTERVIEW METRICS

=====

```
> Saving interview: {"user_id": 9, "domain": "DSA", "score": 85.5, "correct": 17, "total": 20}  
✓ Interview saved: {"session_id": 19, "score": 85.5}
```

=====

TEST 4: GET INTERVIEW HISTORY

=====

```
> Fetching interview history for user: 9  
✓ Retrieved 1 interview(s)
```

Ln 1 Col 1 Spaces: 4 UTF-8 CRLF { } Python 8 venv (3.12.4) (o) Go Live (o) Prettier

PROBLEMS 168 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 168

python test_system.py

```
UniGenAI System Test Suite
    Testing User Management
    Chat Storage & Metrics

=====
TEST 1: USER CREATION
=====
→ Creating user: test_user_1768723549
✓ User created: ID=9, Username=test_user_1768723549

=====
TEST 2: GET USER INFO
=====
→ Fetching user: 9
✓ User retrieved: {"id": 9, "username": "test_user_1768723549", "created_at": "2026-01-18T08:05:51.772956"}

=====
TEST 3: SAVE INTERVIEW METRICS
=====
→ Saving interview: {"user_id": 9, "domain": "DSA", "score": 85.5, "correct": 17, "total": 20}
✓ Interview saved: {"session_id": 19, "score": 85.5}

=====
TEST 4: GET INTERVIEW HISTORY
=====
→ Fetching interview history for user: 9
✓ Retrieved 1 interview(s)
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF { } Python venv (3.12.4) (↻) Go Live ⚡ Prettier

PROBLEMS 168 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 168

python test_system.py

```
✓ Retrieved 1 interview(s)
Sample: {
  "id": 19,
  "domain": "DSA",
  "score": 85.5,
  "correct": 17,
  "total": 20,
  "date": "2026-01-18T08:05:56.243715"
}

=====
TEST 5: GET INTERVIEW STATISTICS
=====
→ Fetching interview stats for user: 9
✓ Stats retrieved successfully
Total Interviews: 1
Average Score: 85.5
Last Score: 85.5
Improvement: 0.0
By Domain: {'DSA': 85.5}

=====
TEST 6: MULTIPLE INTERVIEWS & IMPROVEMENT
=====
✓ Saved OS interview (score: 70.0)
✓ Saved OS interview (score: 75.0)
✓ Saved OS interview (score: 82.5)
✓ Saved DBMS interview (score: 80.0)
✓ Saved DBMS interview (score: 85.0)
✓ Final Stats:
  Total Interviews: 6
  Average Score: 79.67
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF { } Python venv (3.12.4) (↻) Go Live ⚡ Prettier

Auto Switch Working

The screenshot shows a terminal window in VS Code with the title bar "UniGenAI". The terminal tab is active, showing the command `python test_auto_switch.py` being run in a virtual environment. The output displays a test suite named "AUTO-SWITCH TEST SUITE" running three tests. Test 1 and Test 2 both fail with a message indicating a wrong route, while Test 3 succeeds.

```
PS C:\Users\sonal\OneDrive\Desktop\UniGenAI> python test_auto_switch.py
=====
AUTO-SWITCH TEST SUITE
=====

Testing all 4 agents' auto-switch capabilities...

Test 1: [NO_SELECT] → [academic]
Message: 'start mock interview'
X Wrong route! Expected: academic, Got: None

Test 2: [NO_SELECT] → [academic]
Message: 'What is data structure?'
X Wrong route! Expected: academic, Got: None

Test 3: [NO_SELECT] → [academic]
Message: 'Create study plan'
```