
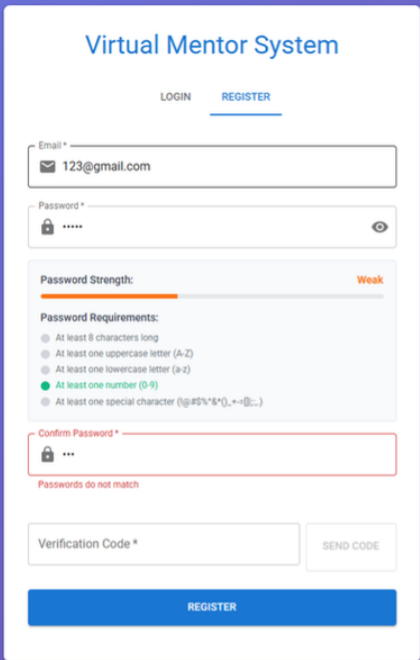


# System test report

## 1. User registration and login

### 1.1 User Registration

- **Method:**
  - a. Open the registration page and fill in a valid email, strong password, and matching confirmation password.
  - b. Click “Send Verification Code” and complete email verification.
  - c. Submit the registration form.
- **Expected Result:**
  - If the input is valid and the verification code is correct, return 200 OK and redirect to the login page with pre-filled email.
  - If the input is invalid or the code is incorrect, show clear error messages near fields or at the top of the form.
- **Result:**
-  Passed. Registration flow works correctly. Verification code is throttled as expected, error messages are accurate, and redirection occurs after success



Virtual Mentor System

LOGIN REGISTER

Email \*  
123@gmail.com

Password \*  
.....

Password Strength: Weak

Password Requirements:

- At least 8 characters long
- At least one uppercase letter (A-Z)
- At least one lowercase letter (a-z)
- At least one number (0-9)
- At least one special character (@#%&'\*()\_+~!@:;)

Confirm Password \*  
...

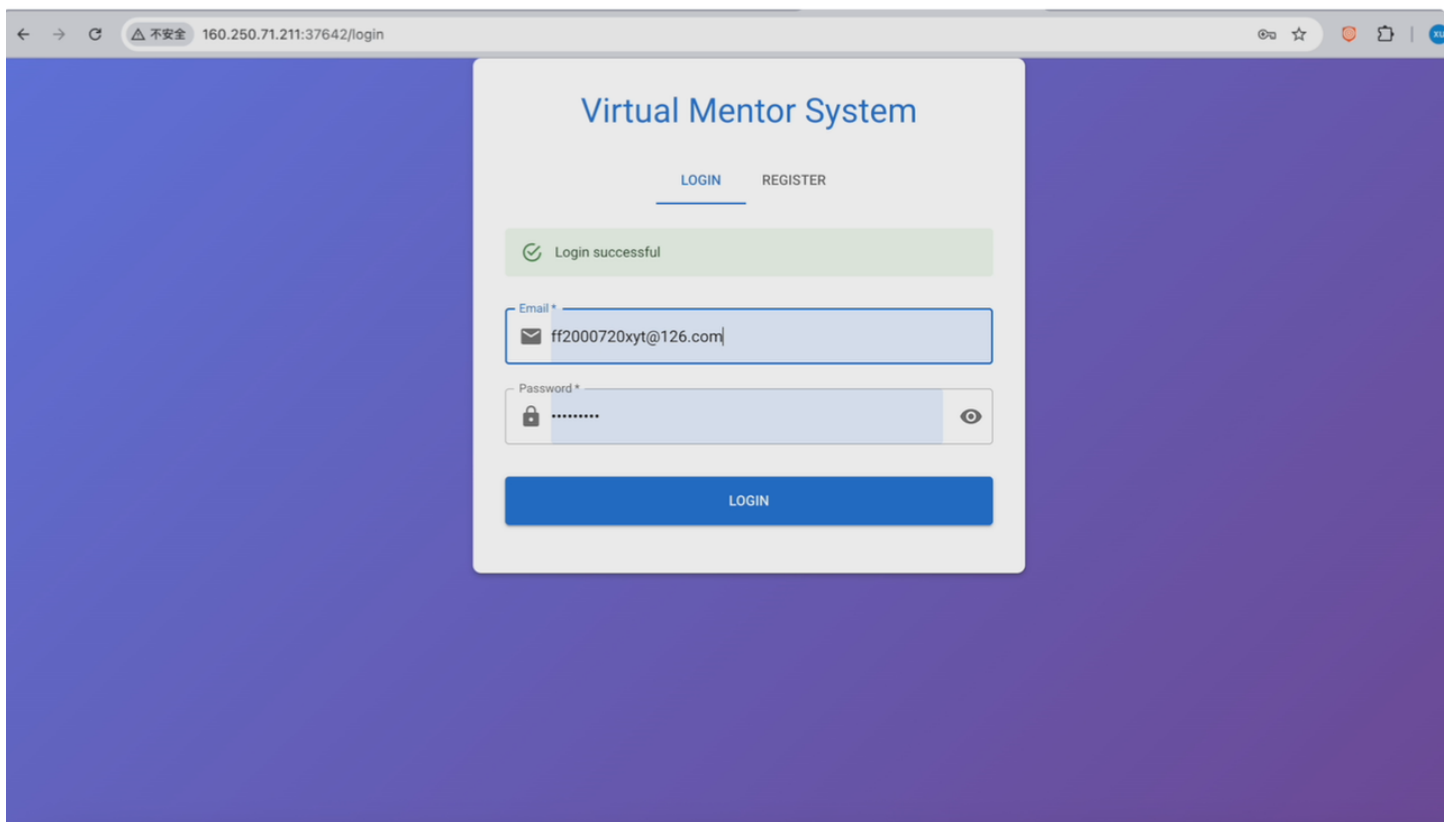
Passwords do not match

Verification Code \* SEND CODE

REGISTER

## 1.2 User Login

- **Method:**
  - a. Visit the login page and enter email and password.
  - b. Click the “Login” button or press Enter to submit.
- **Expected Result:**
  - With correct credentials, return 200 OK with token and user info; redirect based on user role.
  - With incorrect credentials, return 401 Unauthorized and show “invalid email or password” message.
- **Result:**
- ☒ Passed. Redirection works based on role (student → chat page, mentor → admin page). Invalid input shows correct error messages, and session handling is consistent



## 2. User side

### 2.1 User Profile Editing


- **Method:**

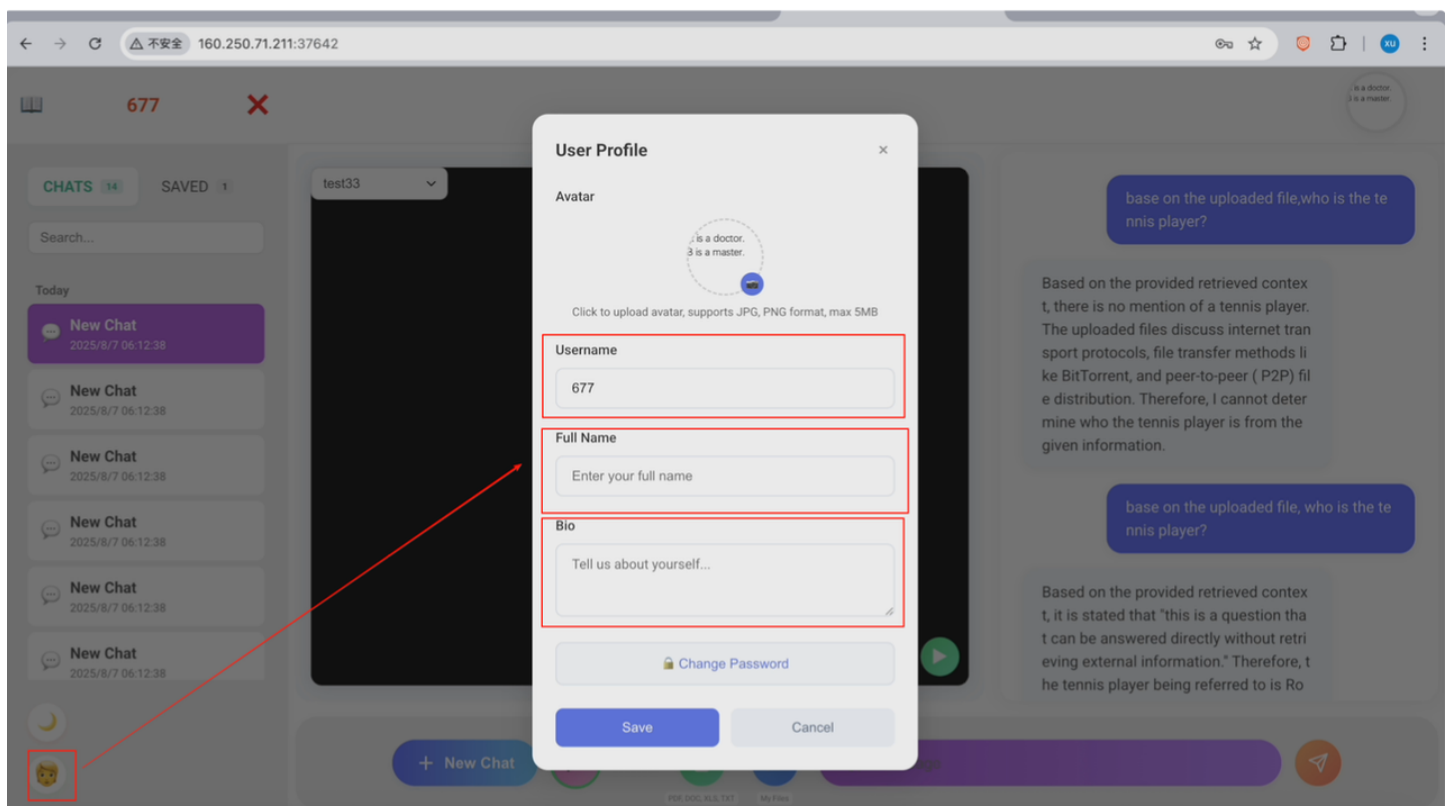
- a. Click the profile icon in the sidebar to open the user settings modal.
- b. Edit username, full\_name, or bio, then click “Save” .

- **Expected Result:**

- On success, update user data locally and display success message.
- On failure, show error message and rollback to previous values.

- **Result:**

-  Passed. Frontend validation and backend syncing work correctly. Errors are handled gracefully



## 2.2 Load Chat Session List

- **Method:**

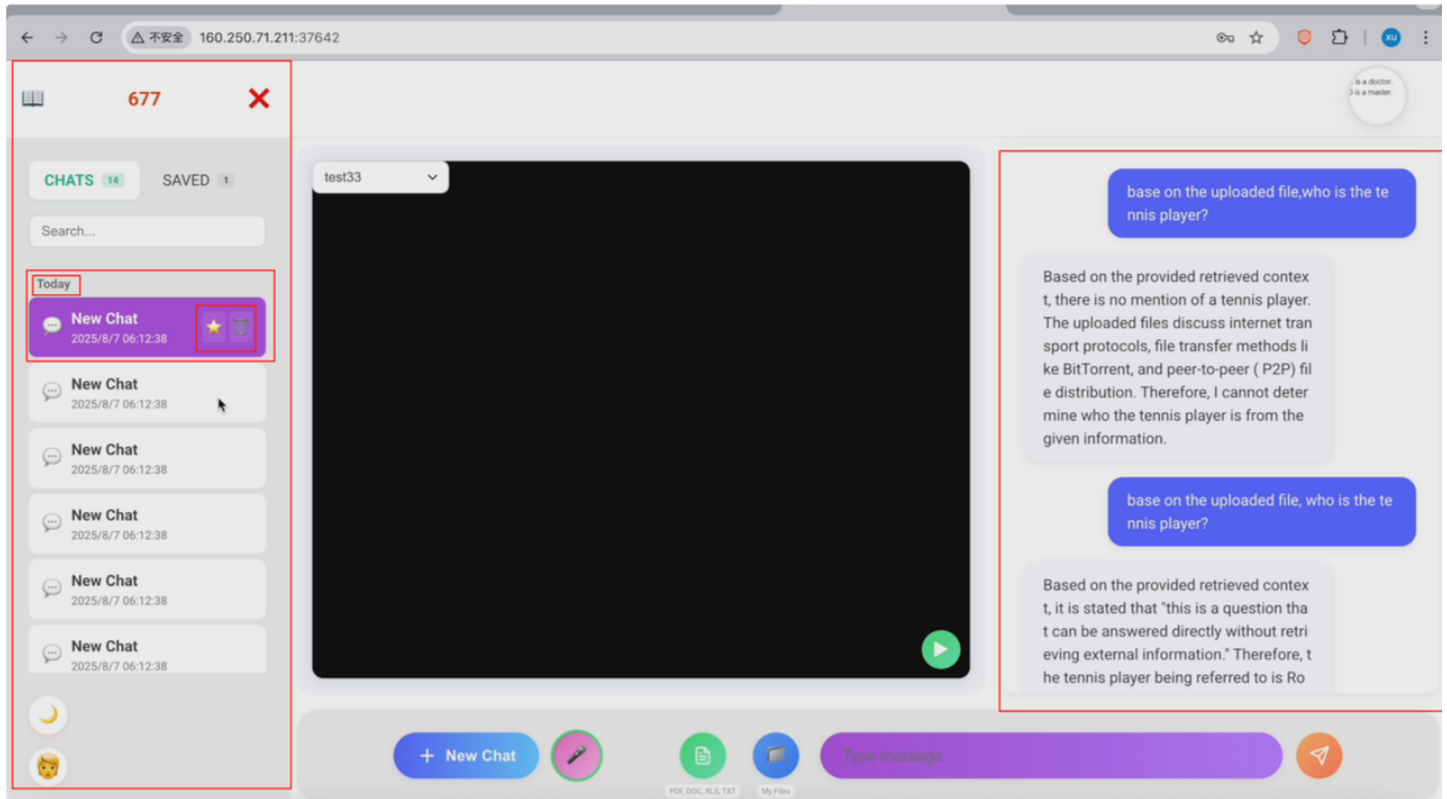
- a. After login as a student, enter the chat page.
- b. The frontend sends a GET /api/chat request with authorization.

- **Expected Result:**

- Return 200 OK with a list of previous conversations.
- Each item should include a timestamp.

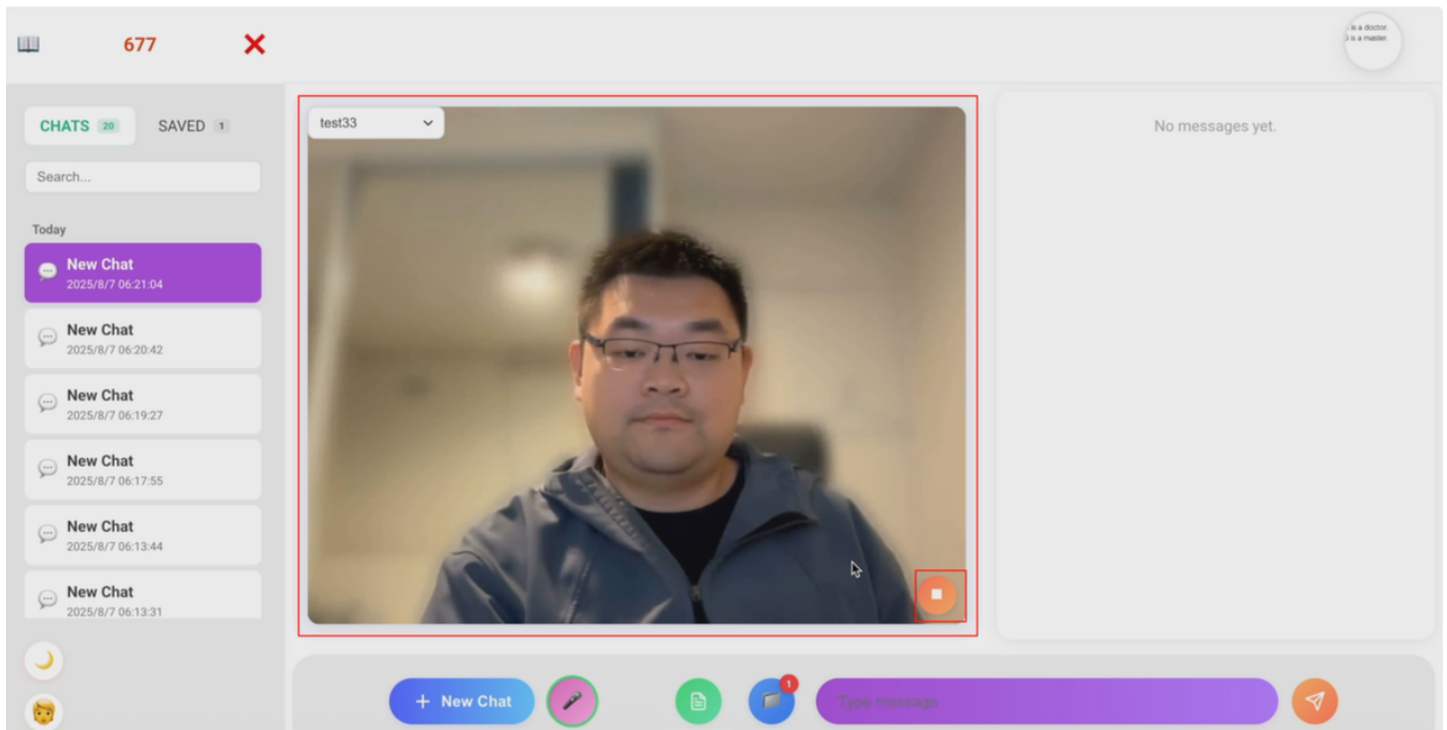
- **Result:**

- ☒ Passed. Chat list loads correctly; list scrolls and updates dynamically. Authorization handled via token in headers



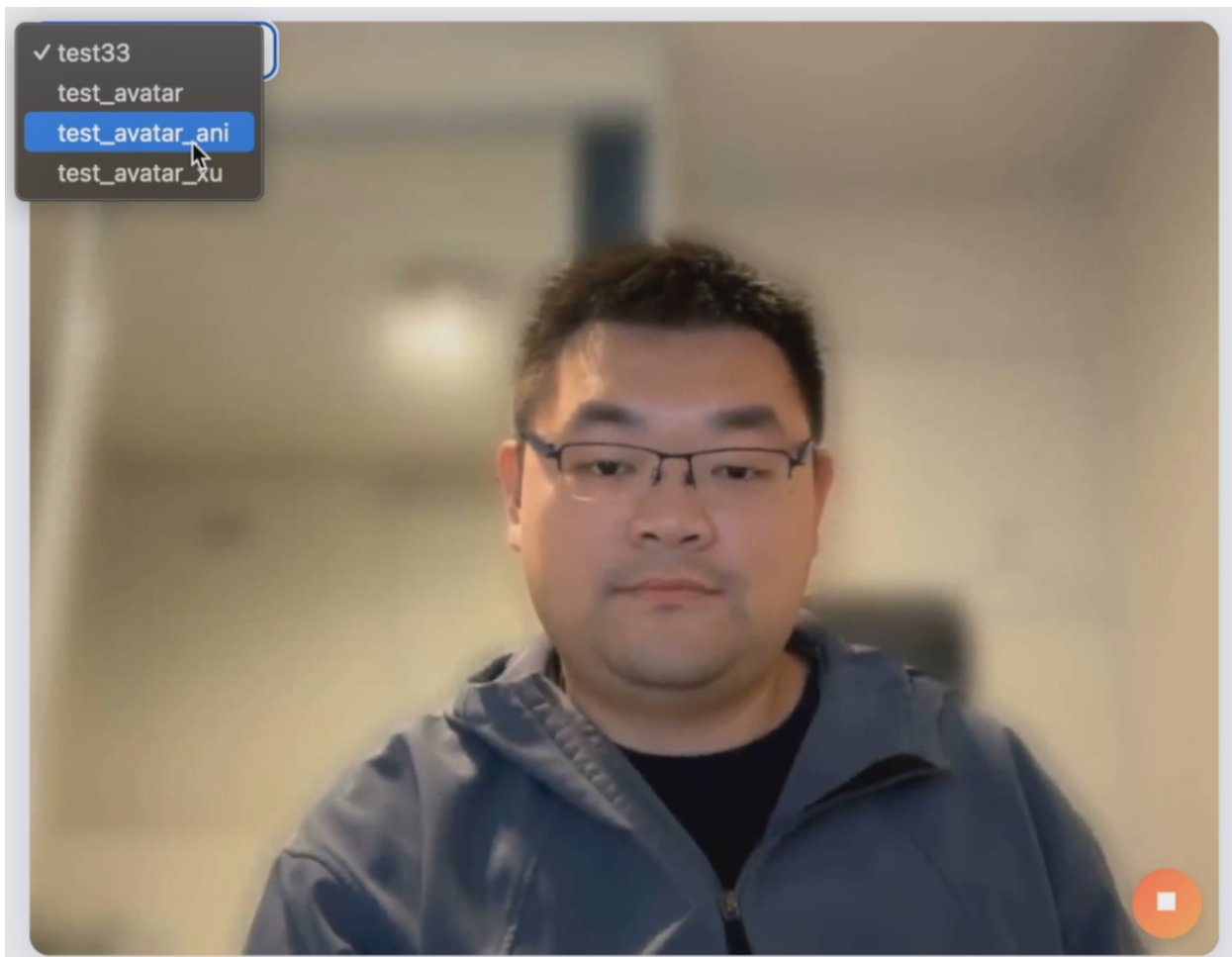
## 2.3 Video Call Setup via WebRTC

- **Method:**
  - a. Click the video icon in the chat window.
  - b. Frontend initiates a WebRTC connection via signaling server (POST /rtc/init).
- **Expected Result:**
  - Both parties should receive peer connection requests.
  - Video stream appears if permission granted.
- **Result:**
- ☒ Passed. 1-to-1 video stream establishes within 2s. ICE negotiation is successful under NAT with TURN fallback.



## 2.4 Avatar Model Switching

- **Method:**
  - a. In the upper right corner of the chat page, click the avatar selection drop-down box.
  - b. Load all available avatars by sending GET `/api/avatar/list`.
  - c. Select one avatar, which triggers a POST `/api/avatar/start` request to activate the associated TTS and lip-sync models.
- **Expected Result:**
  - Avatar list loads successfully, with each item displaying name, image, and language tag.
  - After selecting an avatar, system switches to the corresponding talking head and displays it.
  - Errors in loading or model switching should be handled gracefully.
- **Result:**
- ☒ Passed. Avatar switching is responsive and stable. Video preview updates correctly. Invalid or deleted avatars are filtered out by backend and do not appear in the list



## 2.5 Chat Window Rendering and Interaction

- **Method:**

- a. Open an existing conversation and scroll through past messages.
- b. Send a new message; observe the pending bubble appear immediately.
- c. When the backend responds, verify that the pending message is replaced by the user/tutor message pair.

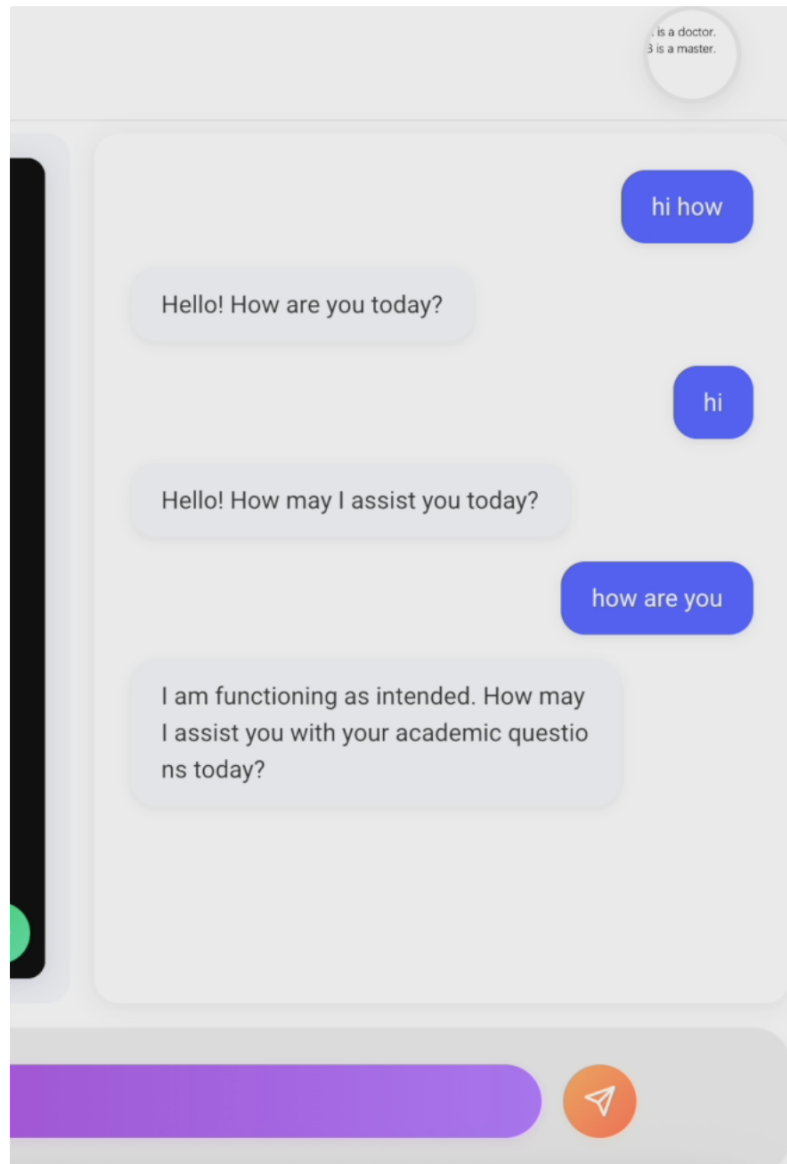
- **Expected Results:**

- Message Alignment: Messages are aligned left and right based on the speaker role (user/tutor).
- Pending Processing: Insert a temporary pending bubble immediately after sending; upon receiving a reply, merge it with the reply into a chronological message pair to maintain visual continuity.

- **Results:**

✓ Passed.

- The pending bubble appears immediately upon sending.
- The merged message pair is rendered in the correct order and alignment after the backend responds.

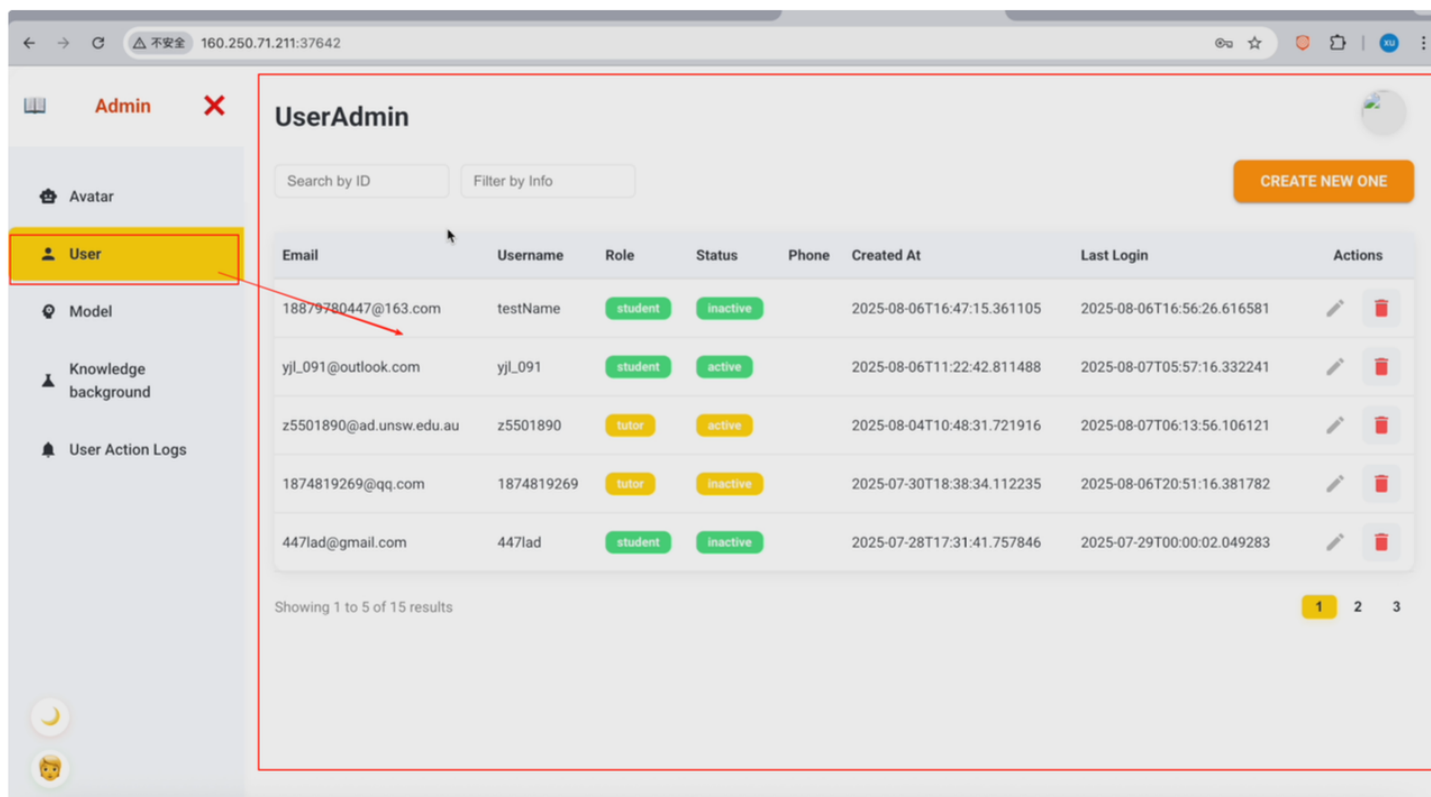


## 3. Administrator side


### 3.1 Load User List

- **Method:**
  - a. Ensure multiple users exist in the database.
  - b. Send a GET request to `/api/admin/users?page=1`.
- **Expected Result:**
  - Return 200 OK with paginated user list.
  - Each user entry includes email, username, role, status.

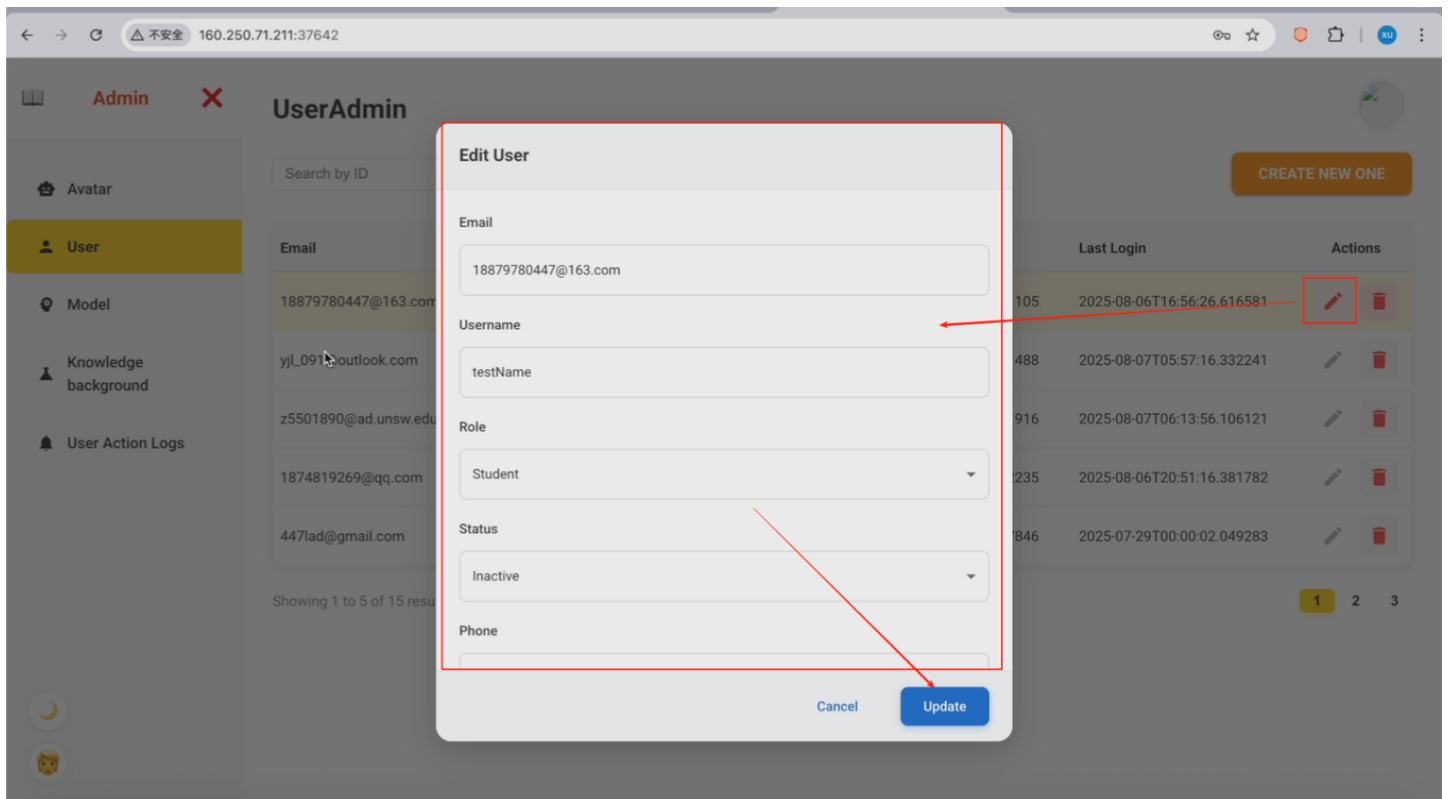
- **Result:**  Success, list renders correctly with skeleton screen while loading.



## 3.2 Edit User Info

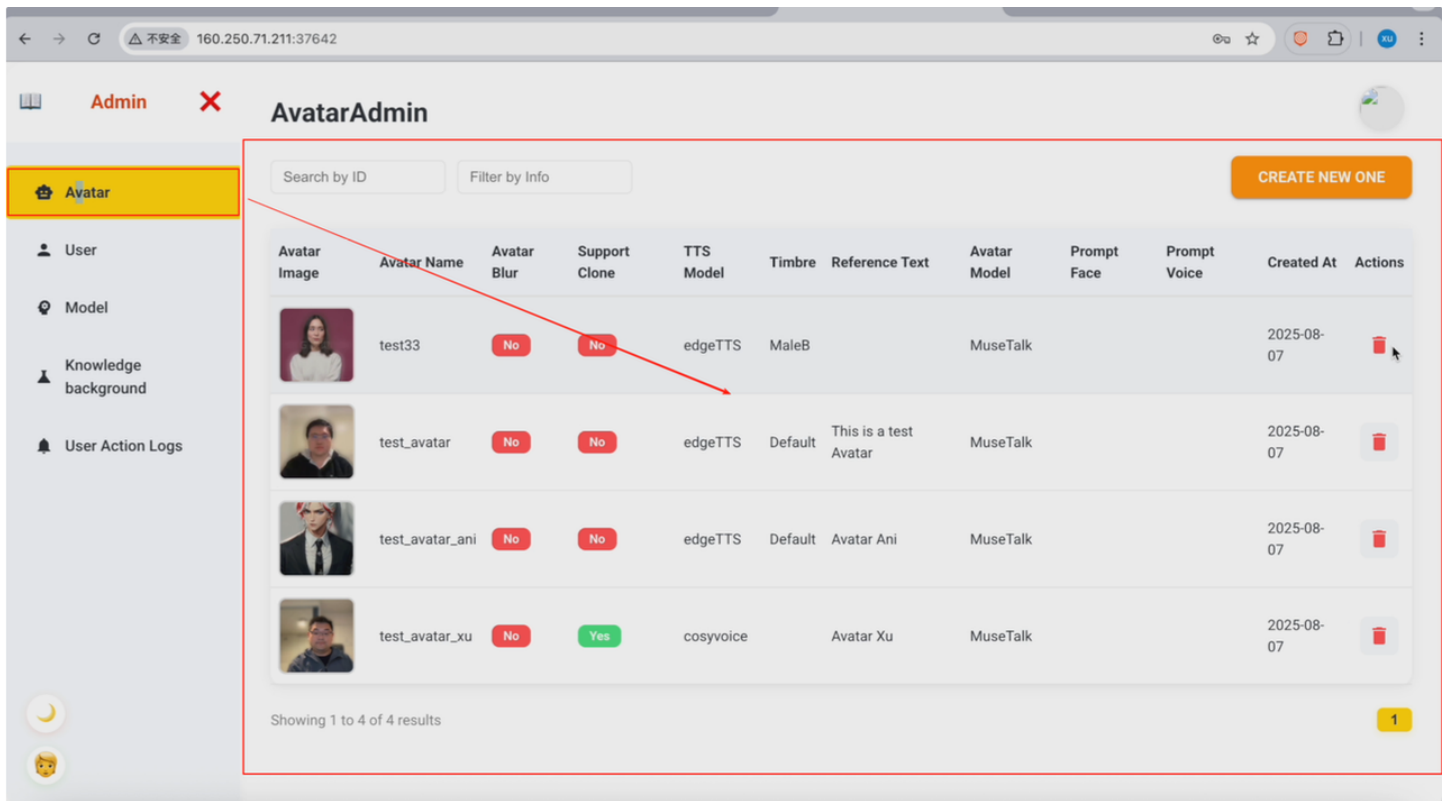
- **Method:**
  - a. Click “Edit” on a user row in the table.
  - b. Modify username or role and click Submit.
- **Expected Result:**
  - Show loading state while submitting.
  - Return 200 OK with updated user info.
  - Display success toast and refresh list.
- **Result:**  Updated successfully and reflected immediately.





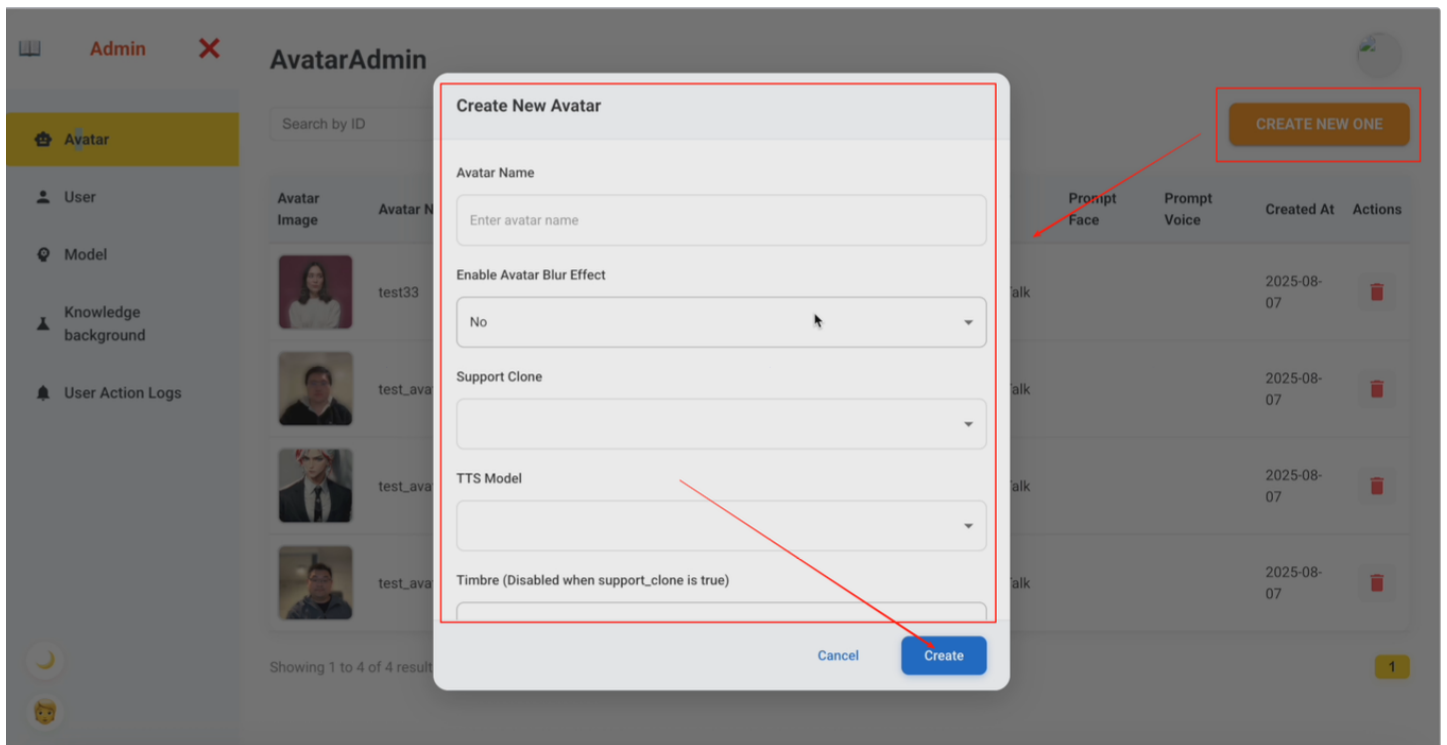
### 3.3 Load Avatar List

- **Method:**
  - a. Ensure several avatars with thumbnail images exist.
  - b. Visit Avatar management page.
- **Expected Result:**
  - Return 200 OK with avatar metadata.
  - Blob preview of avatar thumbnails displays properly.
- **Result:** ☒ Avatar list with thumbnails loads correctly.



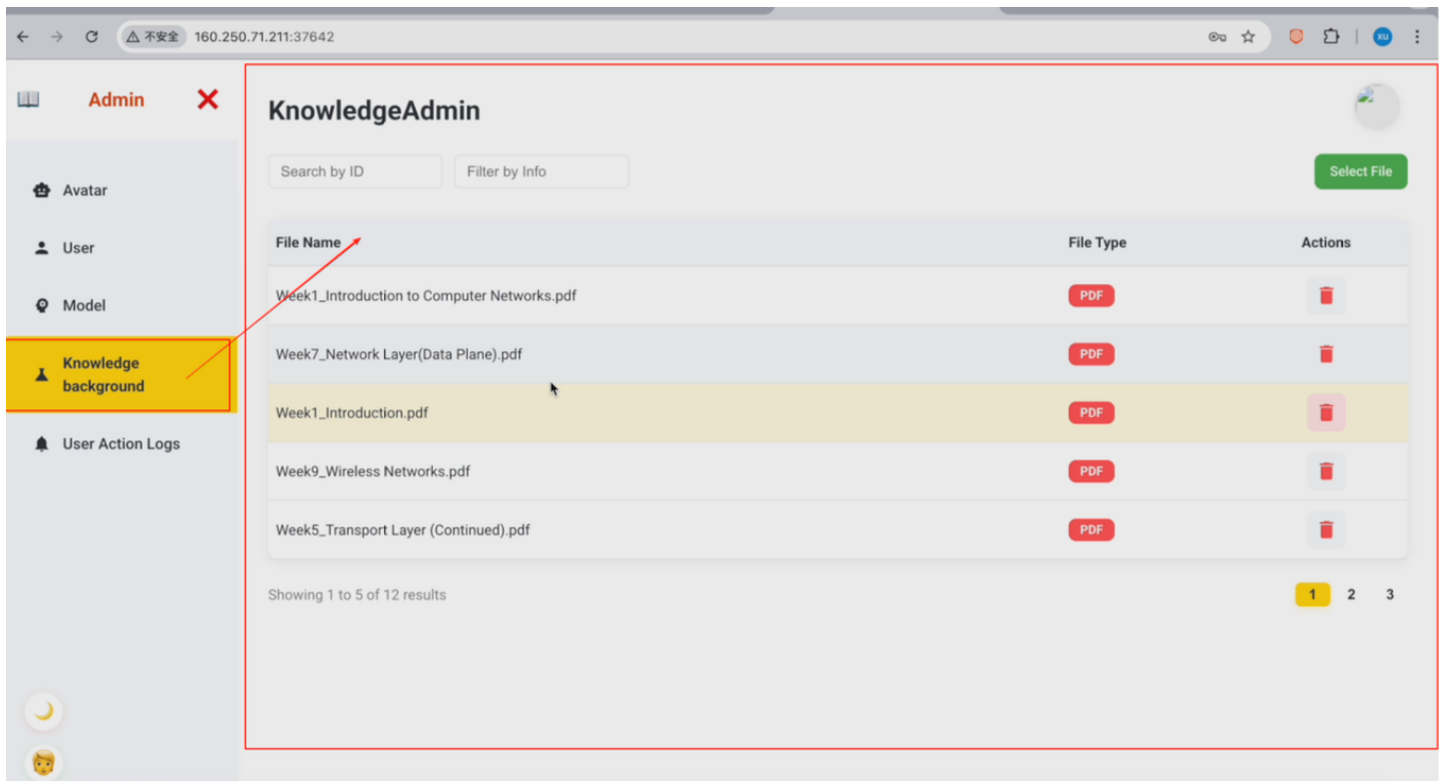
### 3.4 Create New Avatar

- **Method:**
  - a. Click “Create Avatar” .
  - b. Fill in metadata and upload audio/model files.
- **Expected Result:**
  - Verify the file type.
  - Display the animation of the talking head being created.
  - Return 200 on success.
- **Result:** Files validated and uploaded; new avatar appears in list.




### 3.5 Load Public Knowledge Base Files List

- **Method:**
  - a. Ensure public knowledge base files exist on the backend.
  - b. Open the Knowledge Base tab.
- **Expected Result:**
  - Return 200 OK with file metadata.
  - Displays all uploaded knowledge base files.
- **Result:** ☒ Files listed with colored tags and metadata.



### 3.6 Upload New File

- **Method:**
  - a. Click “Select File” , upload a valid document.
  - b. Submit the upload.
- **Expected Result:**
  - Validate file type and size.
  - Return 200 Created.
  - Newly uploaded file appears at top of list.
- **Result:**  Upload validated, file added to top of list.

个人收藏  
下载  
最近使用  
应用程序  
文稿  
xuyuntao  
screenshot  
桌面  
iCloud  
共享  
iCloud 云盘  
位置  
MacBook...  
Macintos...  
OneDrive  
标签  
黄色

下载

今天

大小

种类

添加日期

Course Outline & Logistics.pdf1.8 MBPDF 文稿今天 下午 12:56

COMP9900Final.pptx10.6 MBOffice...L 演示文稿今天 下午 2:38

document\_to\_upload.pdf10 KBPDF 文稿今天 下午 3:19

昨天

新录音 28.m4a32 KBApple...EQ-4 音频昨天 下午 3:37

ani.mp41.9 MBMPEG-4 影片昨天 下午 4:40

过去 7 天

API 需求文档.md.docx229 KBOffice 字处理文稿2025 年 7 月 31 日 下午 3:59

API 需求文档.md (1).docx232 KBOffice 字处理文稿2025 年 7 月 31 日 下午 3:59

API 需求文档.md (2).docx232 KBOffice 字处理文稿2025 年 7 月 31 日 下午 4:00

Week 1 视频播放技术调研.pdf876 KBPDF 文稿2025 年 8 月 2 日 下午 5:57

create\_avatar.py8 KBPython 脚本2025 年 8 月 2 日 下午 7:20

output\_25fps.mp41.3 MBMPEG-4 影片2025 年 8 月 2 日 下午 7:31

取消打开

Select File

Actions

1 2 3