# YÊU CẦU: Chuyển đổi code Puter.js sang Backend Proxy

## 🎯 MỤC TIÊU

Loại bỏ việc users bị chuyển sang puter.com để xác thực. Users sẽ chat trực tiếp mà KHÔNG cần đăng nhập Puter.

---

## PHẦN 1: TẠO BACKEND PROXY (Python Flask)

### 1.1. Thêm Dependencies vào `requirements.txt`

```

flask-cors==4.0.0

requests==2.31.0

```

### 1.2. Tạo file mới hoặc thêm vào Flask hiện có

\*\*Imports cần thiết:\*\*

```python

from flask import Flask, request, jsonify

from flask\_cors import CORS

import requests

import os

```

\*\*Config (đọc từ Replit Secrets):\*\*

```python

PUTER\_USERNAME = os.getenv("PUTER\_USERNAME", "")

PUTER\_PASSWORD = os.getenv("PUTER\_PASSWORD", "")

puter\_token = None

```

\*\*Hàm xác thực backend (chạy ngầm):\*\*

```python

def get\_puter\_token():

"""Xác thực với Puter API - chỉ chạy trên backend"""

global puter\_token

url = "https://api.puter.com/auth/sign-in"

payload = {

"username": PUTER\_USERNAME,

"password": PUTER\_PASSWORD

}

try:

response = requests.post(url, json=payload)

data = response.json()

if response.status\_code == 200 and "token" in data:

puter\_token = data["token"]

print("✅ Backend đã xác thực Puter thành công!")

return True

else:

print(f"❌ Lỗi xác thực: {data}")

return False

except Exception as e:

print(f"❌ Lỗi kết nối: {e}")

return False

```

\*\*API Endpoint cho frontend:\*\*

```python

@app.route('/api/puter-chat', methods=['POST'])

def puter\_chat():

"""Proxy API - Frontend gọi endpoint này thay vì puter.ai.chat()"""

global puter\_token

# Kiểm tra token

if not puter\_token:

if not get\_puter\_token():

return jsonify({"error": "Không thể xác thực Puter"}), 500

# Lấy dữ liệu từ frontend

data = request.get\_json()

message = data.get('message', '')

model = data.get('model', 'gpt-4')

if not message:

return jsonify({"error": "Thiếu message"}), 400

# Map model sang driver

driver\_map = {

'gpt-4': 'openai-completion',

'gpt-3.5-turbo': 'openai-completion',

'claude-3-opus': 'anthropic-completion',

'claude-3-sonnet': 'anthropic-completion',

'llama-3-70b': 'together-completion',

'mixtral-8x7b': 'together-completion'

}

driver = driver\_map.get(model, 'openai-completion')

# Gọi Puter API

url = "https://api.puter.com/drivers/call"

headers = {

"Content-Type": "application/json",

"Authorization": f"Bearer {puter\_token}"

}

payload = {

"interface": "puter-chat-completion",

"driver": driver,

"method": "complete",

"args": {

"messages": [{"role": "user", "content": message}],

"model": model

}

}

try:

response = requests.post(url, json=payload, headers=headers)

# Nếu token hết hạn, lấy lại và retry

if response.status\_code == 401:

print("Token hết hạn, đang lấy lại...")

if get\_puter\_token():

headers["Authorization"] = f"Bearer {puter\_token}"

response = requests.post(url, json=payload, headers=headers)

result = response.json()

# Trả về response

if "message" in result and "content" in result["message"]:

return jsonify({

"reply": result["message"]["content"],

"model": model

})

else:

return jsonify(result)

except Exception as e:

return jsonify({"error": str(e)}), 500

```

\*\*Enable CORS (sau dòng app = Flask(\_\_name\_\_)):\*\*

```python

CORS(app) # Cho phép frontend gọi API

```

\*\*Khởi động server:\*\*

```python

if \_\_name\_\_ == '\_\_main\_\_':

print("🚀 Đang khởi động server...")

get\_puter\_token() # Xác thực 1 lần khi start

port = int(os.getenv("PORT", 5000))

app.run(host='0.0.0.0', port=port)

```

---

## PHẦN 2: SỬA CODE FRONTEND

### 2.1. XÓA hoặc COMMENT OUT các dòng sau:

\*\*Tìm và xóa:\*\*

```html

<!-- XÓA dòng import Puter.js -->

<script src="https://js.puter.com/v2/"></script>

<!-- Hoặc -->

<script>

import puter from 'puter';

</script>

```

```javascript

// XÓA tất cả dòng xác thực

await puter.auth.signIn();

puter.auth.signUp();

puter.auth.getUser();

// XÓA tất cả dòng gọi AI trực tiếp

await puter.ai.chat(...);

await puter.ai.txt2img(...);

```

### 2.2. THAY THẾ bằng code mới:

\*\*Tìm function gọi AI cũ (có thể tên khác nhau):\*\*

```javascript

// CODE CŨ - TÌM VÀ THAY THẾ

async function oldChatFunction(message) {

const response = await puter.ai.chat(message);

return response;

}

```

\*\*THAY BẰNG:\*\*

```javascript

// CODE MỚI - Gọi backend thay vì Puter trực tiếp

async function chatWithAI(message, model = 'gpt-4') {

try {

const response = await fetch('/api/puter-chat', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({

message: message,

model: model

})

});

if (!response.ok) {

throw new Error('Network response was not ok');

}

const data = await response.json();

if (data.error) {

throw new Error(data.error);

}

return data.reply;

} catch (error) {

console.error('Lỗi khi gọi AI:', error);

return 'Đã có lỗi xảy ra: ' + error.message;

}

}

// Sử dụng:

const reply = await chatWithAI('Xin chào!', 'gpt-4');

console.log(reply);

```

### 2.3. Ví dụ cụ thể với button click:

\*\*CODE CŨ:\*\*

```javascript

document.getElementById('sendBtn').addEventListener('click', async () => {

const message = document.getElementById('userInput').value;

// CŨ - Gọi trực tiếp Puter

const response = await puter.ai.chat(message);

displayMessage(response);

});

```

\*\*CODE MỚI:\*\*

```javascript

document.getElementById('sendBtn').addEventListener('click', async () => {

const message = document.getElementById('userInput').value;

const model = document.getElementById('modelSelect').value || 'gpt-4';

// MỚI - Gọi qua backend

const response = await fetch('/api/puter-chat', {

method: 'POST',

headers: { 'Content-Type': 'application/json' },

body: JSON.stringify({ message, model })

});

const data = await response.json();

displayMessage(data.reply);

});

```

---

## PHẦN 3: SETUP CONFIG

```

Key: PUTER\_USERNAME

Value: [username tài khoản puter.com của bạn]

Key: PUTER\_PASSWORD

Value: [password tài khoản puter.com của bạn]

```

---

## PHẦN 4: KIỂM TRA

### ✅ Checklist:

- [ ] Backend có route `/api/puter-chat`

- [ ] Frontend đã xóa tất cả `puter.auth.\*`

- [ ] Frontend đã thay `puter.ai.chat()` bằng `fetch('/api/puter-chat')`

- [ ] Đã thêm vào config: PUTER\_USERNAME và PUTER\_PASSWORD

- [ ] flask-cors đã được cài (trong requirements.txt)

- [ ] CORS đã enable trong Flask app

### 🧪 Test:

1. Chạy server

2. Mở web

3. Gửi tin nhắn chat

4. \*\*KHÔNG BỊ popup puter.com\*\* = Thành công! ✅

---

## YÊU CẦU QUAN TRỌNG

✅ \*\*GIỮ NGUYÊN:\*\*

- Toàn bộ UI/UX hiện tại

- Styling và layout

- Các chức năng khác của web

✅ \*\*CHỈ THAY ĐỔI:\*\*

- Xóa phần xác thực Puter ở frontend

- Thay cách gọi AI: `puter.ai.chat()` → `fetch('/api/puter-chat')`

- Thêm backend proxy

✅ \*\*KẾT QUẢ:\*\*

- Users chat trực tiếp, KHÔNG bị redirect

- Backend tự động xác thực (users không biết)

- Trải nghiệm mượt mà, không có popup

---

## 🆘 NẾU GẶP LỖI

\*\*Lỗi CORS:\*\*

```python

# Đảm bảo có dòng này

CORS(app)

```

\*\*Lỗi 401 Unauthorized:\*\*

- Kiểm tra PUTER\_USERNAME và PUTER\_PASSWORD trong Secrets

- Thử đăng nhập thủ công vào puter.com để verify credentials

\*\*Frontend vẫn bị redirect:\*\*

- Tìm kiếm toàn bộ code với từ khóa: `puter.auth`

- Xóa hoặc comment tất cả