```python
from fastapi import FastAPI, BackgroundTasks, HTTPException
from fastapi.responses import HTMLResponse
import threading
import time
import string
import secrets
from fastapi import Request

"""Simple test for 16x2 character lcd connected to an MCP23008 I2C LCD backpack."""
import board
import adafruit_character_lcd.character_lcd_i2c as character_lcd


# Modify this if you have a different sized Character LCD
lcd_columns = 16
lcd_rows = 4

# Initialise I2C bus
i2c = board.I2C()  # uses board.SCL and board.SDA
# Initialise the lcd class
lcd = character_lcd.Character_LCD_I2C(i2c, lcd_columns, lcd_rows)


app = FastAPI()
code=None
# Function to generate a new code every 60 seconds
def generate_code():
    global code
    while True:

        alphabet = string.digits
        code = ''.join(secrets.choice(alphabet) for i in range(4))
        time.sleep(60)

# Start the code generation in a separate thread
t = threading.Thread(target=generate_code)
t.start()
def get_text(a,b):
    lcd.clear()
    user_string=a
    code=b
    # Turn backlight on
    lcd.backlight = True
    #Print a two line message
    lcd.message = user_string
```

```
46
47      print(user_string)
48
49  # HTML response with a form that includes a text box for user input
50  @app.get("/", response_class=HTMLResponse)
51  async def get_code():
52      return """
53          <h1>Current Code: {code}</h1>
54          <form method="post" action="/">
55              <input type="text" name="user_text">
56              <input type="submit" value="Submit">
57          </form>
58      """.format(code=code)
59
60  # API endpoint to allow users to update the code
61  @app.post("/")
62  async def process_text(request: Request):
63      form_data = await request.form()
64      user_text = form_data.get("user_text")
65      if user_text:
66
67          # Do something with user_text, such as passing it to a function for further
            processing
68          result = get_text(user_text,code)        #return {"result": result}
69      else:
70          raise HTTPException(status_code=400, detail="No text provided")
71
72
73  #async def update_code(background_tasks: BackgroundTasks, new_code: str):
74      #global code
75      #code = new_code
76      #return {"message": f"Message Sent: {new_code}"}
77
78
79
80
```