AI Chatbot - Quick Reference Card

One-Page Cheat Sheet for Students

🚀 Getting Started (5 Minutes)

1. Install Python

```
# Check if installed
python --version

# Should show: Python 3.x.x
```

2. Install Dependencies

```
pip install -r requirements.txt
```

3. Add API Key

Create `.env` file:

```
OPENAI_API_KEY=your-key-here
```

4. Run App

```
streamlit run app.py
```

Visit: `http://localhost:8501`

📁 File Structure

```
Project/
├── app.py              # Main application
├── config.py           # Settings & API key
├── school_data.py      # School information
├── requirements.txt    # Dependencies
└── .env               # API key (secret!)
```

🐍 Python Essentials

Variables & Types

```
# String
name = "Alice"

# Number
age = 15
score = 95.5

# List
grades = [90, 85, 95]

# Dictionary
student = {"name": "Alice", "grade": 10}
```

Functions

```
def greet(name):
    return f"Hello, {name}!"

result = greet("Alice")  # "Hello, Alice!"
```

If/Else

```
if score >= 90:
    grade = "A"
elif score >= 80:
    grade = "B"
else:
    grade = "C"
```

Loops

```
# For loop
for item in my_list:
    print(item)

# While loop
while count < 10:
    count += 1
```

🎨 Streamlit Basics

Display Text

```
st.title("My App")
st.header("Section")
st.write("Hello, World!")
```

Get Input

```
name = st.text_input("Enter name:")
age = st.number_input("Enter age:")
choice = st.selectbox("Choose:", ["A", "B", "C"])
```

Buttons

```
if st.button("Click Me"):
    st.write("Button clicked!")
```

Forms

```
with st.form("myform"):
    text = st.text_input("Input:")
    submit = st.form_submit_button("Submit")

    if submit:
        st.write(f"You entered: {text}")
```

Session State (Remember Data)

```
# Initialize
if "count" not in st.session_state:
    st.session_state.count = 0

# Use
st.session_state.count += 1
st.write(st.session_state.count)
```

Layout

```
# Sidebar
with st.sidebar:
    st.write("In sidebar")
```

```
# Columns
col1, col2 = st.columns(2)
with col1:
    st.write("Column 1")
with col2:
    st.write("Column 2")
```

🤖 OpenAI API

Basic Usage

```
import openai

client = openai.OpenAI(api_key="sk-...")

response = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[
        {"role": "system", "content": "You are helpful"},
        {"role": "user", "content": "Hello!"}
    ]
)

answer = response.choices[0].message.content
```

Message Roles

- **system**: Instructions for AI
- **user**: Human's message
- **assistant**: AI's response

Important Parameters

```
model="gpt-4o-mini"     # Which AI to use
max_tokens=1000          # Response length
temperature=0.7          # Creativity (0-1)
```

🔧 Common Patterns

Retry with Error Handling

```
for attempt in range(3):
    try:
        result = api_call()
```

```
            break
        except Exception as e:
            if attempt < 2:
                time.sleep(1)
                continue
            else:
                return f"Error: {e}"
```

Save/Load Data

```
import pickle

# Save
with open("data.pkl", "wb") as f:
    pickle.dump(my_data, f)

# Load
with open("data.pkl", "rb") as f:
    my_data = pickle.load(f)
```

Environment Variables

```
import os
from dotenv import load_dotenv

load_dotenv()
api_key = os.getenv("OPENAI_API_KEY")
```

🐛 Troubleshooting

Problem: "Module not found"

```
pip install <module-name>
```

Problem: "API key not found"

1. Check `.env` file exists
2. Check key starts with `sk-`
3. Restart the app

Problem: "Messages not saving"

```
# Check initialization
if "messages" not in st.session_state:
```

```
    st.session_state.messages = []
```

Problem: "Form not clearing"

```
# Add clear_on_submit=True
with st.form("form", clear_on_submit=True):
    ...
```

📊 Project Structure Tips

Separate Concerns

- **config.py**: Settings only
- **school_data.py**: Data only
- **app.py**: Application logic

Use Constants

```
# Bad
if len(msg) > 1000:
    ...

# Good
MAX_LENGTH = 1000
if len(msg) > MAX_LENGTH:
    ...
```

Document Functions

```
def calculate_total(a, b):
    """
    Add two numbers together

    Args:
        a (int): First number
        b (int): Second number

    Returns:
        int: Sum of a and b
    """
    return a + b
```

🔒 Security Checklist

- ✅ Never commit API keys to Git
- ✅ Add `.env` to `.gitignore`
- ✅ Use environment variables
- ✅ Validate user input
- ✅ Handle errors gracefully

📚 Key Concepts

Token: Unit of text AI processes (≈4 chars)

Session State: Streamlit's way to remember data

API: Way for programs to talk to each other

Prompt: Instructions you give the AI

Temperature: AI creativity level (0=focused, 1=creative)

🎯 Quick Commands

Terminal

```
# Install package
pip install package-name

# Run Streamlit
streamlit run app.py

# Check Python version
python --version

# Create virtual environment
python -m venv venv

# Activate virtual env (Mac/Linux)
source venv/bin/activate

# Activate virtual env (Windows)
venv\Scripts\activate
```

Git

```
# Initialize repository
git init

# Add files
git add .

# Commit
```

```
git commit -m "Description"

# Push to GitHub
git push origin main
```

💡 Best Practices

1. **Start Simple**: Build basic version first
2. **Test Often**: Run app after each change
3. **Read Errors**: They tell you what's wrong
4. **Use Comments**: Explain complex code
5. **Save Frequently**: Commit to Git often
6. **Ask for Help**: Use Stack Overflow, forums

🌟 Next Steps

Beginner Projects

- Customize school info
- Change colors/styling
- Add new quick actions

Intermediate Projects

- Add user authentication
- Create analytics dashboard
- Multi-user support

Advanced Projects

- Database integration
- Voice input/output
- Real-time notifications

📖 Resources

Documentation:

- Streamlit: docs.streamlit.io
- OpenAI: platform.openai.com/docs
- Python: docs.python.org

Learning:

- Python Tutorial: python.org/tutorial
- Real Python: realpython.com
- freeCodeCamp: freecodecamp.org

Community:

- Stack Overflow: stackoverflow.com
- Streamlit Forum: discuss.streamlit.io
- GitHub: github.com

🆘 Common Errors & Fixes

Error	Fix
`ModuleNotFoundError`	`pip install <module>`
`KeyError: 'messages'`	Initialize session state
`API key invalid`	Check `.env` file
`Connection timeout`	Check internet, retry
`Form not clearing`	Add `clear_on_submit=True`
`Page not refreshing`	Use `st.rerun()`

🎓 Success Tips

✅ Read error messages carefully ✅ Start with working examples ✅ Change one thing at a time ✅ Use print() to debug ✅ Google error messages ✅ Don't give up - errors are learning!

This project teaches you:

- Python programming
- Web development
- AI integration
- API usage
- Data management
- Problem solving

Keep building! Every error is progress! 🚀

For complete guide, see COMPLETE_LEARNING_GUIDE.pdf

Version 1.0 | October 2025