In following the guide on full-stack I've copied over the chart that's provided.
API:

Endpoint	Method	What is does
GET /api/status	GET	Health check - Returns {"status": "ok"}
GET /api/messages	GET	Get all chat messages
POST /api/chat	POST	Send a message, get AI reply
DELETE /api/messages/<id>	DELETE	Delete a specific message
GET /api/health	GET	Detailed health info (database, version, metrics) (i feel like this could be labeled as 'version' rather than 'health')
GET /api/metrics	GET	Request count and performance data

Defining a Route (URL Pattern)

In python you define a url route with `@app.route("/")`, then right below you will define a function `def home():` for example. There you can add whatever functionality you want when someone access that url.

Using jsonify will return it in a json format

Flask

Start the server with `app.run()`, it will listen on port 5000 for HTTP requests, then Flask will match the URL to the route, Flask will call the matching function, function will return a response (JSON), Flask sends that response back to the client

Linting

run linting often to check your code before its compiled and ran with `ruff check app/`

BACKEND: Step 2

There was an issue with the guide here. If you follow as is there is an error

```
(venv) sethstigall@fedora:~/engrcs_agentic_ai_tutorial/aiproject/backend$ python -m pytest tests/ -v
========================================= test session starts ==========================================
platform linux -- Python 3.14.6, pytest-9.1.1, pluggy-1.6.0 -- /home/sethstigall/engrcs_agentic_ai_tutori
al/aiproject/backend/venv/bin/python
cachedir: .pytest_cache
rootdir: /home/sethstigall/engrcs_agentic_ai_tutorial/aiproject/backend
plugins: anyio-4.14.2, flask-1.3.0
collected 1 item

tests/test_status.py::test_status_returns_ok ERROR                                              [100%]

================================================ ERRORS ================================================
_____________________________ ERROR at setup of test_status_returns_ok _____________________________
file /home/sethstigall/engrcs_agentic_ai_tutorial/aiproject/backend/tests/test_status.py, line 1
  def test_status_returns_ok(client):
file /home/sethstigall/engrcs_agentic_ai_tutorial/aiproject/backend/venv/lib64/python3.14/site-packages/p
ytest_flask/plugin.py, line 111
  @pytest.fixture(autouse=True)
  def _configure_application(request, monkeypatch):
E       fixture 'app' not found
>       available fixtures: _configure_application, _monkeypatch_response_class, _push_request_context, a
ccept_any, accept_json, accept_jsonp, accept_mimetype, anyio_backend, anyio_backend_name, anyio_backend_o
ptions, cache, capfd, capfdbinary, caplog, capsys, capsysbinary, capteesys, client, client_class, config,
 doctest_namespace, free_tcp_port, free_tcp_port_factory, free_udp_port, free_udp_port_factory, live_serv
er, monkeypatch, pytestconfig, record_property, record_testsuite_property, record_xml_attribute, recwarn,
 subtests, tmp_path, tmp_path_factory, tmpdir, tmpdir_factory
>       use 'pytest --fixtures [testpath]' for help on them.

/home/sethstigall/engrcs_agentic_ai_tutorial/aiproject/backend/venv/lib64/python3.14/site-packages/pytest
_flask/plugin.py:111
======================================= short test summary info ========================================
ERROR tests/test_status.py::test_status_returns_ok
========================================== 1 error in 0.01s ============================================
```

Using Claude Code I was able to learn that we need to set up the conftest.py file. There we put the fixture for pytest to call. Here is the file I added:

backend/conftest.py

```
import pytest
from app import create_app


@pytest.fixture
def app():
    app = create_app()
    app.config.update(TESTING=True)
    return app


@pytest.fixture
def client(app):
    return app.test_client()
```

I also needed to update the test itself by changing `assert response.get_json() == {"status": "ok"}` to `assert response.get_json() == {"status": "ok", "version": "1.0"}` to make sure that it matches whats returned from the response

BACKEND: Step 3: Database

There is another error where we are asked to run the tests, but the first one fails. I'm sure it has something to do with us never having included the `@app.route("/api/chat` that `test_create_message(client)` is testing. In the next step (4) we move to using blueprints, so I'll just skip to that version to see if the tests will pass.

BACKEND: Step 4: API

In this step it seems that most of the errors form before were fixed, but the `test_delete_message_works` is failing. Again there still seems to be an issue where the guide is instructing people to use tests that we haven't finished setting up to pass yet.

When we start to manually test API endpoints there is an issue with starting our app:

```
(venv) sethstigall@fedora:~/engrcs_agentic_ai_tutorial/aiproject/backend$ python app/__init__.py
Traceback (most recent call last):
  File "/home/sethstigall/engrcs_agentic_ai_tutorial/aiproject/backend/app/__init__.py", line 27, in <mod
ule>
    app = create_app()
  File "/home/sethstigall/engrcs_agentic_ai_tutorial/aiproject/backend/app/__init__.py", line 17, in crea
te_app
    from .routes import api
ImportError: attempted relative import with no known parent package
```

The issue seems to be in how files are being imported, so Claude recommends that we create a separate entry point to the app with `backend/run.py`;

```
from app import create_app

app = create_app()

if __name__ == "__main__":
     app.run(debug=True, port=5000)
```

MOBILE: Step 3: API Client

I had an error `ERROR [Error: Uncaught (in promise, id: 0): "ReferenceError: Property 'setError' doesn't exist"]`
This was fixed by adding `const [error, setError] = useState<string | null>(null);` to the top of the App section. Inside of App.tsx it looks like:

```
export default function App() {
  const [messages, setMessages] = useState<Message[]>([]);
```

```
  const [input, setInput] = useState("");
  const [error, setError] = useState<string | null>(null);

  const handleSend = async () => {...
```

This gets covered in the next step.

MOBILE: Step 4: State

This is minor, but towards the end it states that you should test that the loading indicator works. This isn't implemented until the next step.

I also found that you need to set loading to true near the top of `handleSend` like this:

```
const handleSend = async () => {
const text = input.trim();
if (!text) return; setInput("");
setError(null);
setLoading(true); //The added part
```

Then after your catch block you add:

```
finally {
  setLoading(false);
}
```

MOBILE: Step 5: Polish

If you had an issue with the "AI is typing..." popping up then make sure you implemented the setLoading(true) -> setLoading(false).

Next there was an issue with getting timestamps to arrive properly. Make sure you define renderMessage in App() and not in the module section. Then you need to update FlatList to has renderItem = renderMessage.

```
<FlatList
  ref={flatListRef}
  data={messages}
  keyExtractor={(item) => item.id.toString()}
  onContentSizeChange={() => flatListRef.current?.scrollToEnd()}
```

```
    renderItem={renderMessage}
    />
```