

Starting with 5usd let's see how far we get. Let's make a simple CRUD app for daily notes with a text field and rating of 1-5. We'll use python and fastapi for the backend and Lynx for a mobile application.

1. Asked claude to create folders for backend and frontend which worked nicely
2. Asking it to set up the project it fails with using "python" but tries again with python3
3. But it doesn't realize it's already in the backend folder and tries to cd again...
 1. Pointing out it's in the folder already and telling to create python env and it gets it right
4. "Implement the following routes /create /update:id /delete:id /alldays"
5. Routes look ok, let's also create unit tests
6. And let's ask Claude how I can run the tests myself
 1. "source venv/bin/activate && python -m pytest test_app.py -v"
7. Tests fail with "AssertionError: non-body parameters must be in path, query, header or cookie"
8. Let's take a look at what we have...
 1. Backend is using plain object for in-memory notes
 2. http methods are ok
 3. error handling is in place (e.g. 404 note not found)
9. I'm not familiar with fastapi so let's ask Claude to explain the create endpoints code where test is failing...explanation is understandable and concise.
 1. fastapi parses the data automatically based on the model defined
 1. `@app.post("/create", response_model=DayNote)`
 2. `"def create_day_note(content: str, rating: int = Field(..., ge=1, le=5)):"`
10. test puts the data in params?
 1. `response = client.post("/create", params={"content": "Test day note", "rating": 4})`
 2. Looking at the error I'd assume we need to use something else then "params", quick googling would point to "json"
11. Let's get Claude to fix this. But first let's commit this so we get a nice history.
 1. Claude edits app.py and tests...
12. Running tests again we fail with incorrect assert `"test_delete_day_note - assert 422 == 404"`
13. Let's get Claude to fix this again and it makes multiple edits...
14. Tests pass! And claude wants to create requirements.txt so let's do that. It also want's to create a Readme so let's add that too.
15. Claude also wants to start the backend so let's do that
 1. Claude starts the backend and hangs itself there waiting for server to die? :D

- Let's start the server to test it out:
- **Bash**(source venv/bin/activate && python app.py)...
- * **Ideating...** (114s · esc to interrupt)

> █

! for bash mode · / for commands · esc to undo

2.

3. Server start dies

```

• Bash(source venv/bin/activate && python app.py)...
  INFO: Will watch for changes in these directories: ['/Users/mattias/learn/claudecodes/backend']
  INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
  INFO: Started reloader process [52260] using StatReload
  INFO: Started server process [52262]
  INFO: Waiting for application startup.
  INFO: Application startup complete.

  Command interrupted
  Exit code 143

```

- The server is running! You can interact with the API using the Swagger UI at <http://localhost:8000/docs>.

> █

! for bash mode · / for commands · esc to undo

4.

5. No, it just took surprisingly long and the server is running and somehow has Swagger docs at <http://localhost:8000/docs> ?

6. Nope, nothing replies from that path as nothing is implemented there.

16. Let's start it ourselves and everything seems to start ok so let's assume the backend is done for now.

```

▶(venv) → backend git:(main) python3 app.py
INFO: Will watch for changes in these directories: ['/Users/mattias/learn/claudecodes/backend']
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
INFO: Started reloader process [52501] using StatReload
INFO: Started server process [52503]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: 127.0.0.1:65051 - "GET / HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:65051 - "GET /favicon.ico HTTP/1.1" 404 Not Found

```

1.