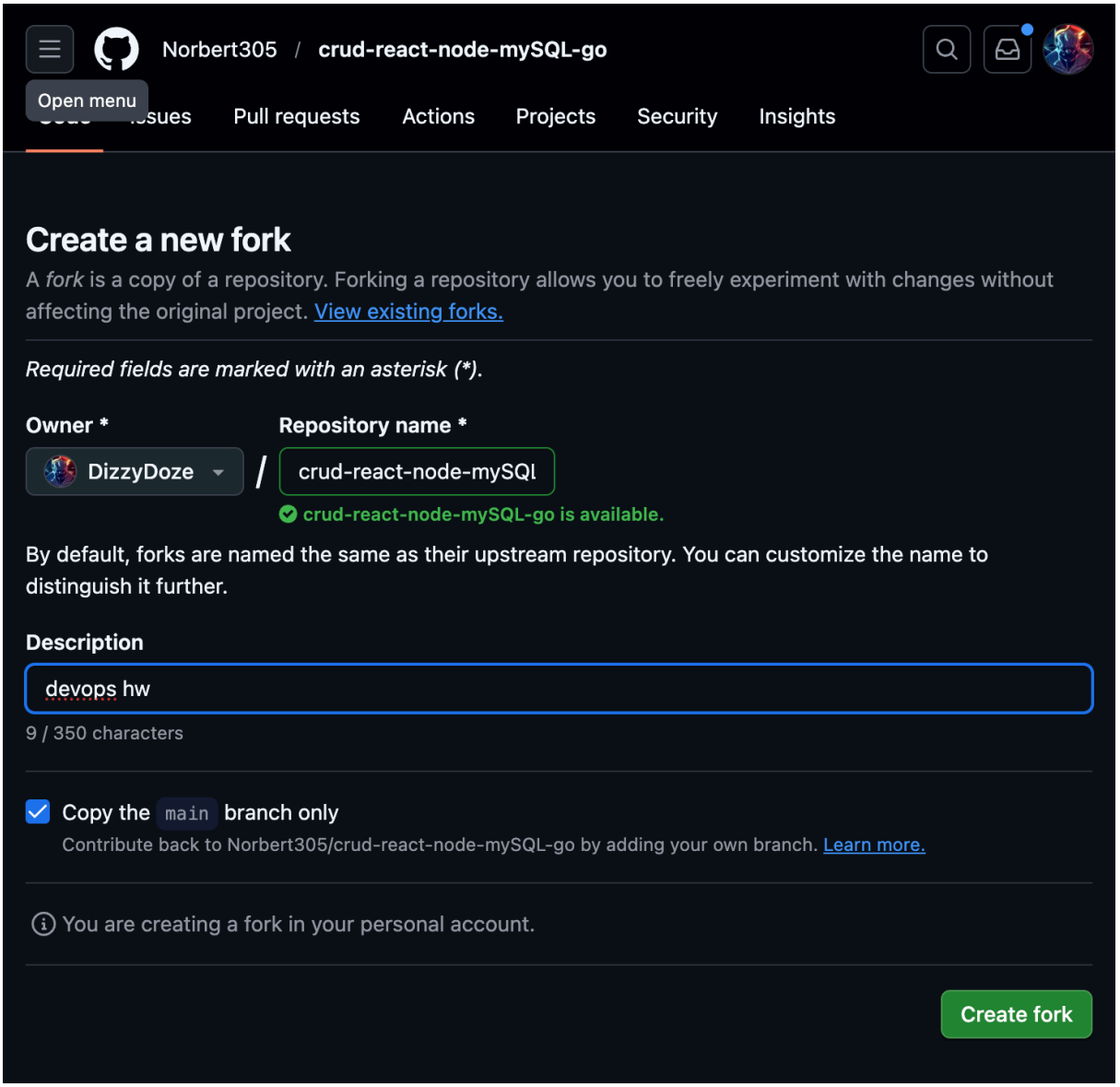
HW4-Containerize it

Repo Address

> https://github.com/DizzyDoze/crud-react-node-mySQL-go

- `git clone https://github.com/DizzyDoze/crud-react-node-mySQL-go`
- `cd crudxxxxx`
- `docker compose up -d –build`
- Open browser and go for http://localhost:4173

Fork the repo & Clone




```
yj@YJs-MacBook-Air DevOps % git clone git@github.com:DizzyDoze/crud-react-node-mySQL-go.git
Cloning into 'crud-react-node-mySQL-go'...
remote: Enumerating objects: 1113, done.
remote: Counting objects: 100% (58/58), done.
remote: Compressing objects: 100% (50/50), done.
remote: Total 1113 (delta 25), reused 8 (delta 8), pack-reused 1055 (from 1)
Receiving objects: 100% (1113/1113), 1.25 MiB | 1.88 MiB/s, done.
Resolving deltas: 100% (164/164), done.
```

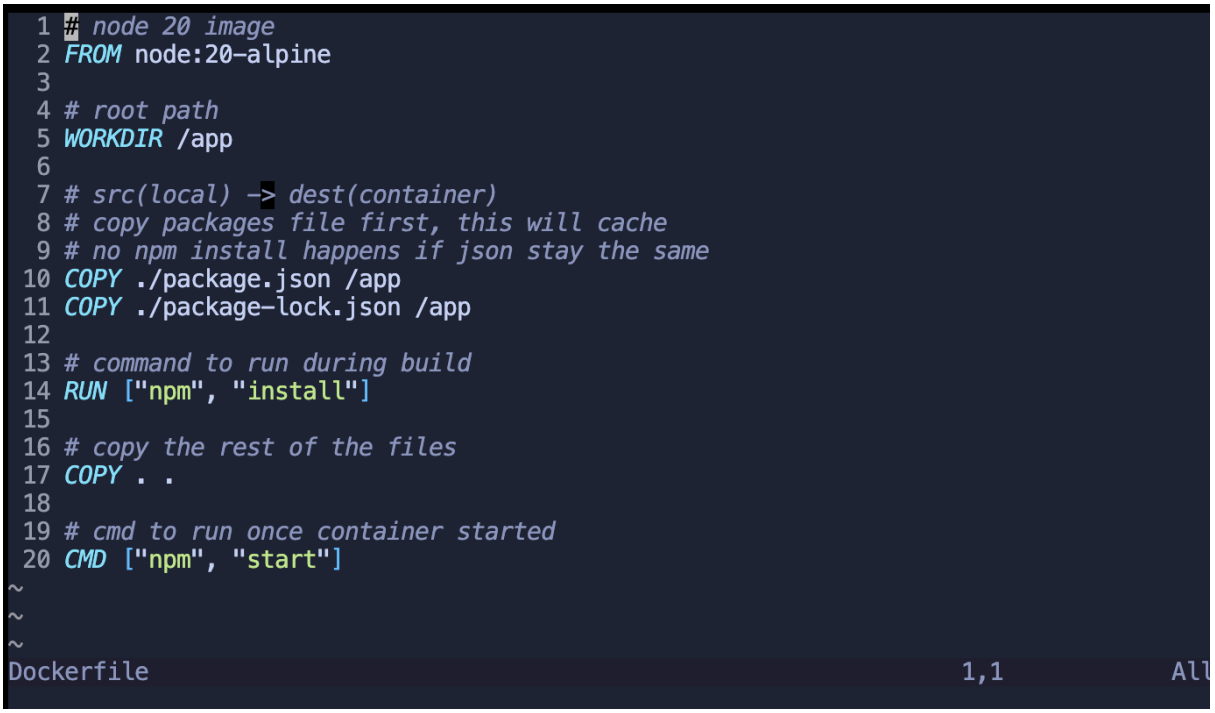
Create Dockerfile for the Backend

Confirm repo tags

- `22-bullseye`, `22.22-bullseye`, `22.22.0-bullseye`, `jod-bullseye`
- `22-bullseye-slim`, `22.22-bullseye-slim`, `22.22.0-bullseye-slim`, `jod-bullseye-slim`
- `22-trixie`, `22.22-trixie`, `22.22.0-trixie`, `jod-trixie`
- `22-trixie-slim`, `22.22-trixie-slim`, `22.22.0-trixie-slim`, `jod-trixie-slim`
- `20-alpine3.22`, `20.20-alpine3.22`, `20.20.0-alpine3.22`, `iron-alpine3.22`
- `20-alpine`, `20-alpine3.23`, `20.20-alpine`, `20.20-alpine3.23`, `20.20.0-alpine`, `20.20.0-alpine3.23`, `iron-alpine`, `iron-alpine3.23`
- `20`, `20-bookworm`, `20.20`, `20.20-bookworm`, `20.20.0`, `20.20.0-bookworm`, `iron`, `iron-bookworm`
- `20-bookworm-slim`, `20-slim`, `20.20-bookworm-slim`, `20.20-slim`, `20.20.0-bookworm-slim`, `20.20.0-slim`, `iron-bookworm-slim`, `iron-slim`
- `20-bullseye`, `20.20-bullseye`, `20.20.0-bullseye`, `iron-bullseye`
- `20-bullseye-slim`, `20.20-bullseye-slim`, `20.20.0-bullseye-slim`, `iron-bullseye-slim`
- `20-trixie`, `20.20-trixie`, `20.20.0-trixie`, `iron-trixie`
- `20-trixie-slim`, `20.20-trixie-slim`, `20.20.0-trixie-slim`, `iron-trixie-slim`

Create Dockerfile

- Copy the npm install dependent files first, they will cache



```
# node 20 image
FROM node:20-alpine

# root path
WORKDIR /app

# src(local) -> dest(container)
# copy packages file first, this will cache
# no npm install happens if json stay the same
COPY ./package.json /app
COPY ./package-lock.json /app

# command to run during build
RUN ["npm", "install"]

# copy the rest of the files
COPY . .

# cmd to run once container started
CMD ["npm", "start"]
~
~
~
Dockerfile                                                    1,1           All
```

- Create .dockerignore to avoid copy the whole module dir

```
node_modules/
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
.dockerignore                                                 1,1           All
```

- PS: Update **mysql2**, as mysql 5+ is no longer provided by the docker hub, we use 9.6, will have auth problem.

```
{
  "name": "backend",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "type": "module",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "nodemon index.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "cors": "^2.8.5",
    "express": "^4.18.2",
    "mysql2": "^3.6.0",
    "nodemon": "^3.0.1"
  }
}
~
package.json                                                15,21         All
```

Create Dockerfile for Frontend

- Check package.json see how to start and build via vite

```
  "version": "0.0.0",
  "type": "module",
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "lint": "eslint . --ext js,jsx --report-unused-disable-directives --max-warnings 0
",
    "preview": "vite preview"
  },
  "dependencies": {
    "axios": "^1.5.0",
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
    "react-router-dom": "^6.16.0"
  },
  "devDependencies": {
    "@types/react": "^18.2.15",
    "@types/react-dom": "^18.2.7",
    "@vitejs/plugin-react": "^4.0.3",
    "eslint": "^8.45.0",
    "eslint-plugin-react": "^7.32.2",
    "eslint-plugin-react-hooks": "^4.6.0",
    "eslint-plugin-react-refresh": "^0.4.3",
    "vite": "^4.4.5"
  }
```

- Create the Dockerfile

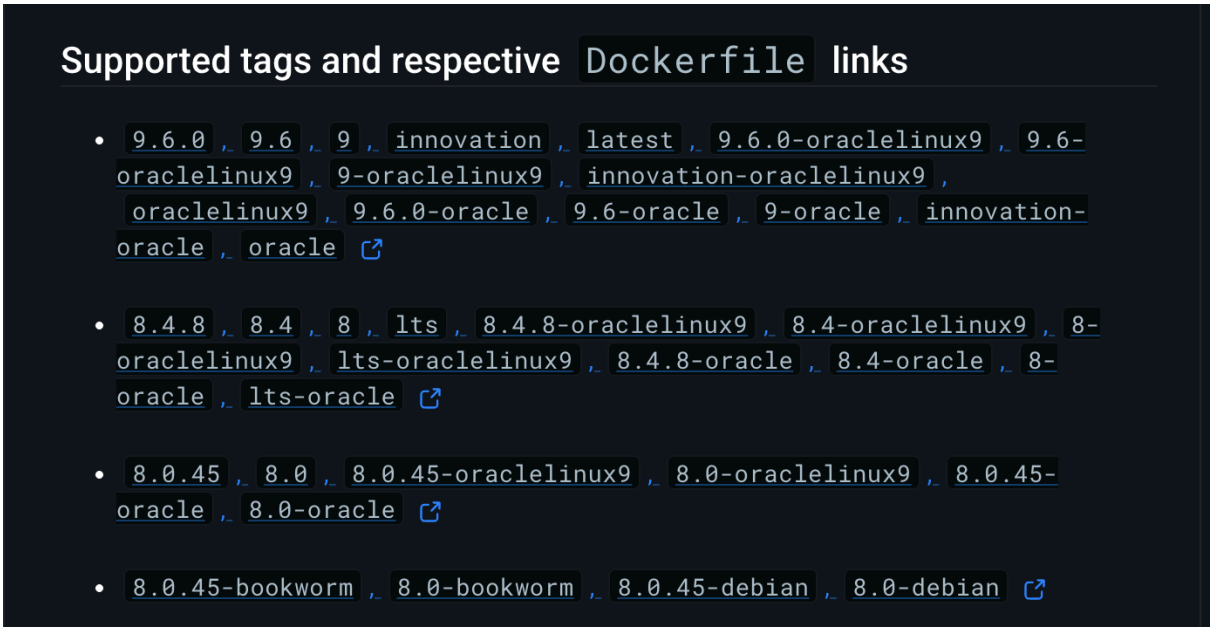
```
# image base
FROM node:20-alpine

# work dir
WORKDIR /app

# COPY the package dependencies, cache, no install if json file stays the same
COPY ./package.json /app
COPY ./package-lock.json /app

# RUN, run during the image build
RUN ["npm", "install"]

# copy the rest
COPY . .

# only one CMD per dockerfile, run after the container started
RUN ["npm", "run", "build"]

# -- means the rest of args are for vite preview, not npm
CMD ["npm", "run", "preview", "--", "--host", "0.0.0.0"]

~
Dockerfile                                                    1,1           All
```

Create Docker-Compose file for all Services

- Check the index.js from backend we know there's **MySQL** service

```
const db = mysql.createConnection({
    host: "localhost",
    user: "root",
    password: "",
    database: "test"
})
```

- Check MySQL images tags



- Create compose file
  - port mapping
  - service name is the container hostname which used to connect from each other
  - **IMPORTANT**: depends_on avoid crashing, wait for db container to start
    - which is NOT enough, need to add **healthcheck**, with **service_healthy** condition, which is another **keyword**, once the healthcheck cmd works, docker compose will then start the backend container.
  - volumes, tell db container to auto run init.db so table will be created

```yaml
services:
  backend:
    build:
      context: backend  # build the image from Dockerfile under backend
    ports:
      - "8800:8800"      # host: container, must be string
    depends_on:
      db:
        condition: service_healthy  # docker keyword, look for healthcheck keyword

  frontend:
    build:
      context: frontend  # build the image from Dockerfile under frontend
    ports:
      - "4173:4173"      # host: container, must be string

  db:
    image: mysql:9.6
    environment:
      MYSQL_ROOT_PASSWORD: admin123
      MYSQL_DATABASE: test
    # map local path init.sql to mysql container path, it will auto run the init.sql
    volumes:
      - ./db/init.sql:/docker-entrypoint-initdb.d/init.sql
    # backend container won't start until the mysql actually connected
    healthcheck:
      test: ["CMD", "mysqladmin", "ping", "-h", "localhost"]
      interval: 5s
      timeout: 3s
      retries: 5

```

Build and Run

```
yj@YJs-MacBook-Air crud-react-node-mySQL-go % docker compose up -d --build
[+] Running 11/11
 ✔ db Pulled                                                              11.6s
   ✔ ad37a81ec0b0 Pull complete                                            0.5s
   ✔ bdaccdb6a2d1 Pull complete                                            7.0s
   ✔ 8bd823d80b30 Pull complete                                            0.5s
   ✔ f3308f5dc5a7 Pull complete                                            0.4s
   ✔ 78c8f56b46ce Pull complete                                            7.6s
   ✔ c74b2b2e44e4 Pull complete                                            0.5s
   ✔ cb8e577a10e0 Pull complete                                           10.0s
   ✔ dbc311b7141a Pull complete                                            0.5s
   ✔ 9b9e9c3d0312 Pull complete                                            7.2s
   ✔ 2866f080bd66 Pull complete                                            0.3s
[+] Building 9.8s (24/24) FINISHED
```

```
[+] Running 6/6
 ✔ crud-react-node-mysql-go-backend               Built                   0.0s
 ✔ crud-react-node-mysql-go-frontend              Built                   0.0s
 ✔ Network crud-react-node-mysql-go_default       Created                 0.0s
 ✔ Container crud-react-node-mysql-go-frontend-1  Started                 0.4s
 ✔ Container crud-react-node-mysql-go-db-1        Started                 0.4s
 ✔ Container crud-react-node-mysql-go-backend-1   Started                 0.3s
yj@YJs-MacBook-Air crud-react-node-mySQL-go %
```

- We have backend, frontend and db

```
yj@YJs-MacBook-Air crud-react-node-mySQL-go % docker ps
CONTAINER ID   IMAGE                          COMMAND                  CREATED         STATUS         PORTS                                       NAMES
b4d32fcdd86c   crud-react-node-mysql-go-backend    "docker-entrypoint.s…"   2 minutes ago   Up 2 minutes   0.0.0.0:8800->8800/tcp, [::]:8800->8800/tcp   crud-react-node-mysql-go-backend-1
4d3c0f564828   crud-react-node-mysql-go-frontend   "docker-entrypoint.s…"   2 minutes ago   Up 2 minutes   0.0.0.0:4173->4173/tcp, [::]:4173->4173/tcp   crud-react-node-mysql-go-frontend-1
83e550f4a5e8   mysql:9.6                      "docker-entrypoint.s…"   2 minutes ago   Up 2 minutes   3306/tcp, 33060/tcp                         crud-react-node-mysql-go-db-1
```

- Backend logs

```
yj@YJs-MacBook-Air crud-react-node-mySQL-go % docker logs -f crud-react-node-mysql-go-backend-1

> backend@1.0.0 start
> nodemon index.js

[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
Connect to the backend!!!!
```

- Frontend logs

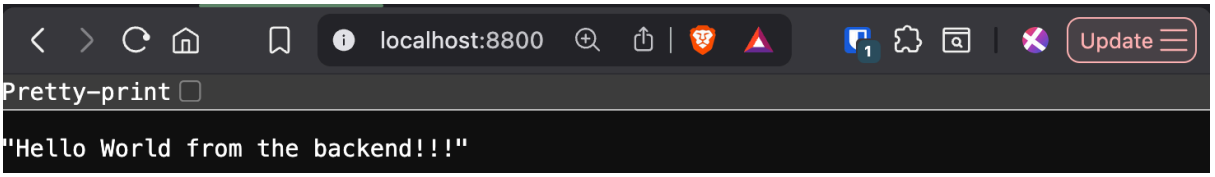
```
yj@YJs-MacBook-Air crud-react-node-mySQL-go % docker logs -f crud-react-node-mysql-go-frontend-1

> frontend@0.0.0 preview
> vite preview --host 0.0.0.0

  ➜  Local:   http://localhost:4173/
  ➜  Network: http://172.19.0.3:4173/
```

- DB logs

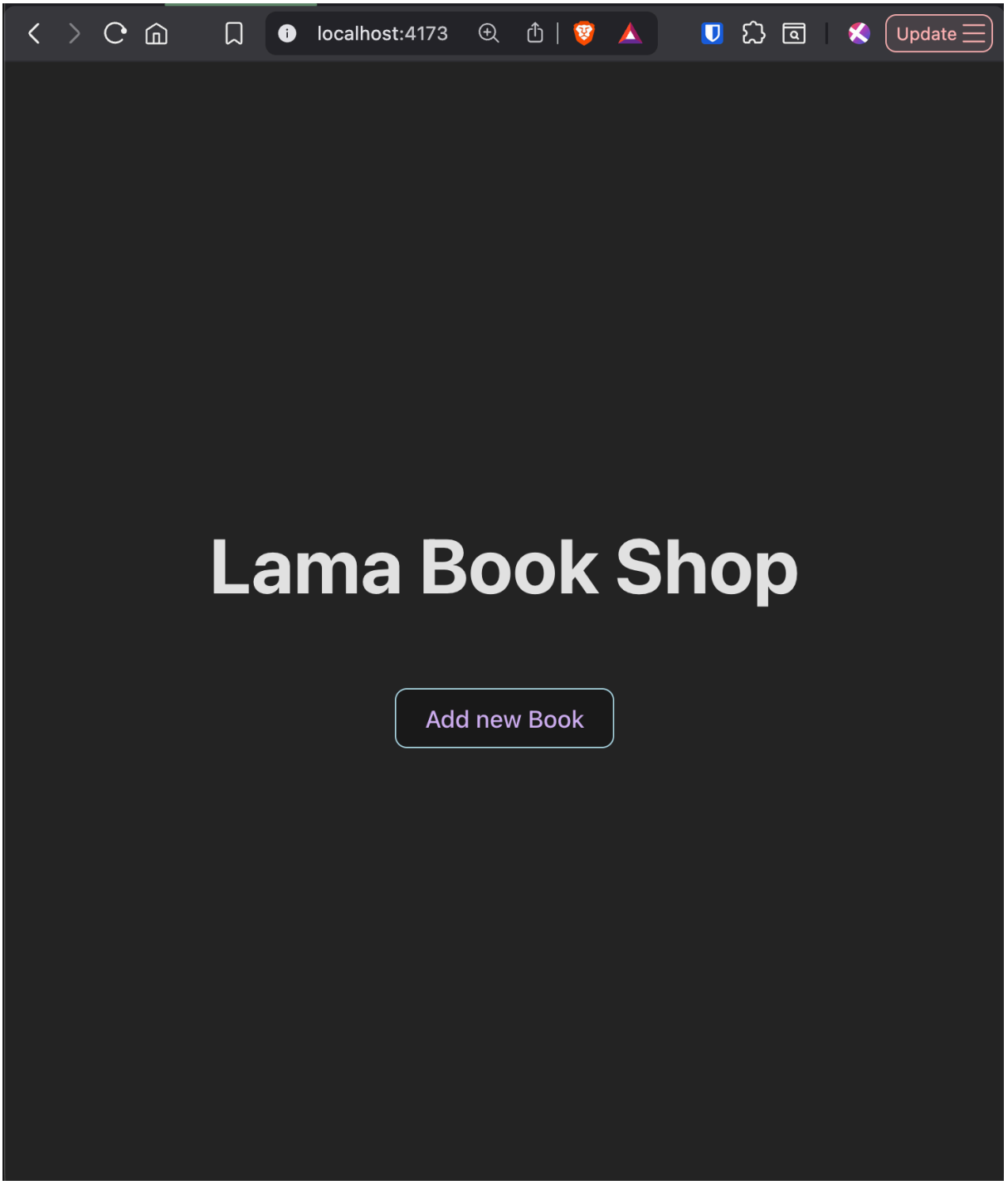
```
2026-02-07T01:41:40.296287Z 0 [System] [MY-015015] [Server] MySQL Server - start.
2026-02-07T01:41:40.445671Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 9.6.0) starting as process 1
2026-02-07T01:41:40.445678Z 0 [System] [MY-015590] [Server] MySQL Server has access to 10 logical CPUs.
2026-02-07T01:41:40.445689Z 0 [System] [MY-015590] [Server] MySQL Server has access to 8218034176 bytes of physical memory.
2026-02-07T01:41:40.448642Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2026-02-07T01:41:40.512937Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2026-02-07T01:41:40.625252Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2026-02-07T01:41:40.625271Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now supported for this ch
nnel.
2026-02-07T01:41:40.626241Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is accessible to a
l OS users. Consider choosing a different directory.
2026-02-07T01:41:40.632735Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::' port: 33060, socket: /var/run/mysqld/mysqlx.so
k
2026-02-07T01:41:40.632763Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '9.6.0'  socket: '/var/run/mysqld/mysqld.sock'
port: 3306  MySQL Community Server - GPL.
```

Test

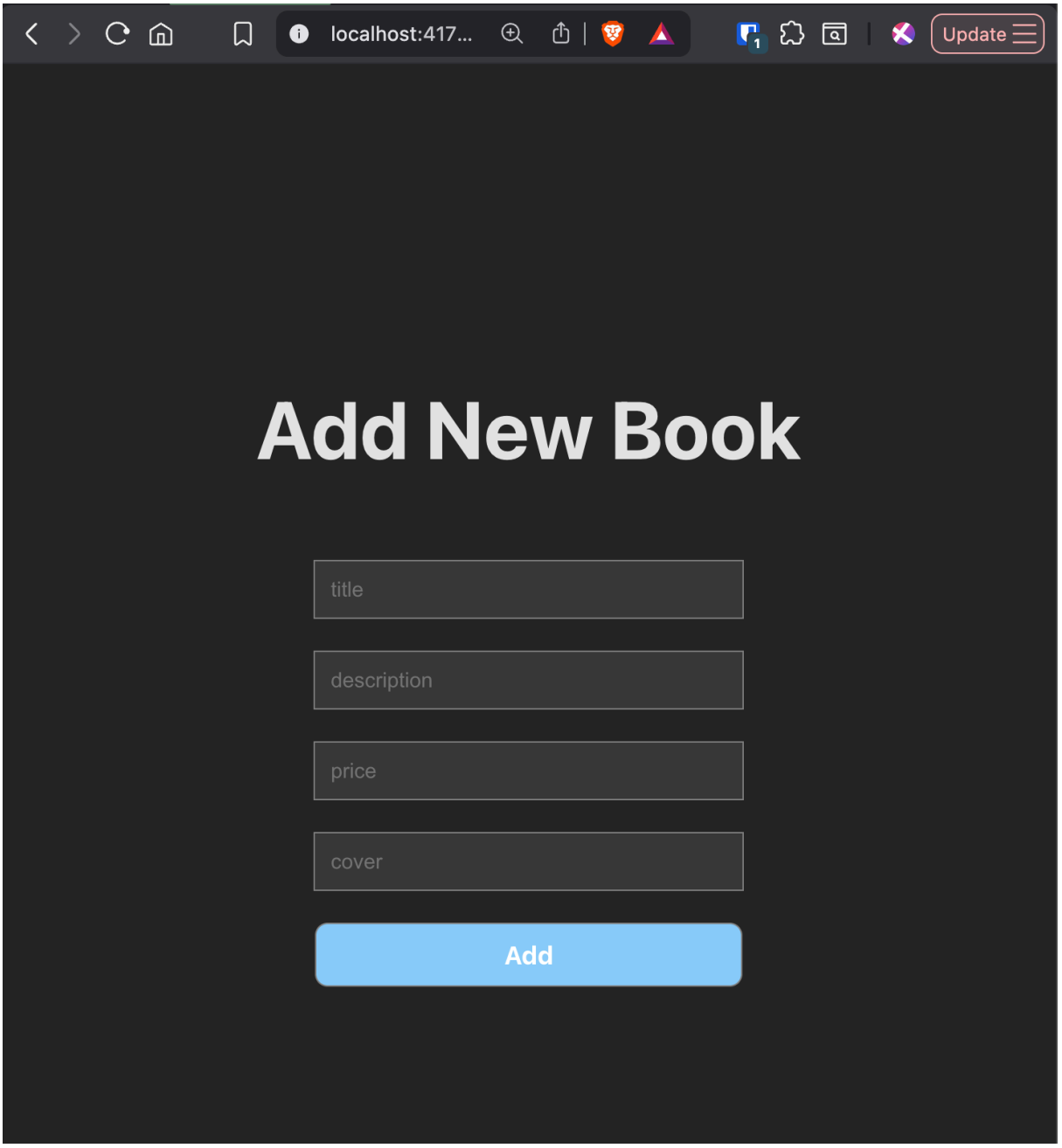
- Backend OK

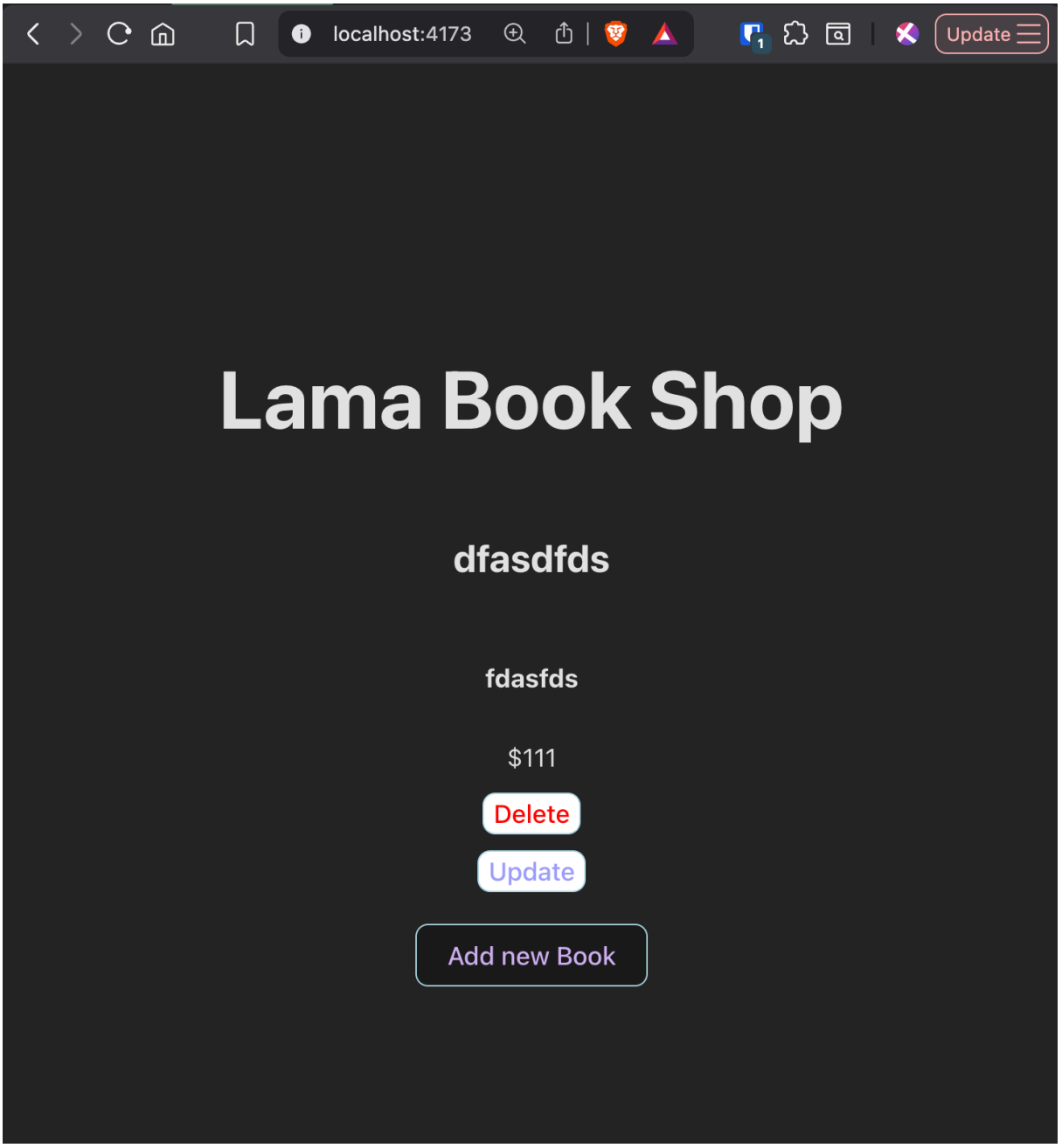


- Frontend OK

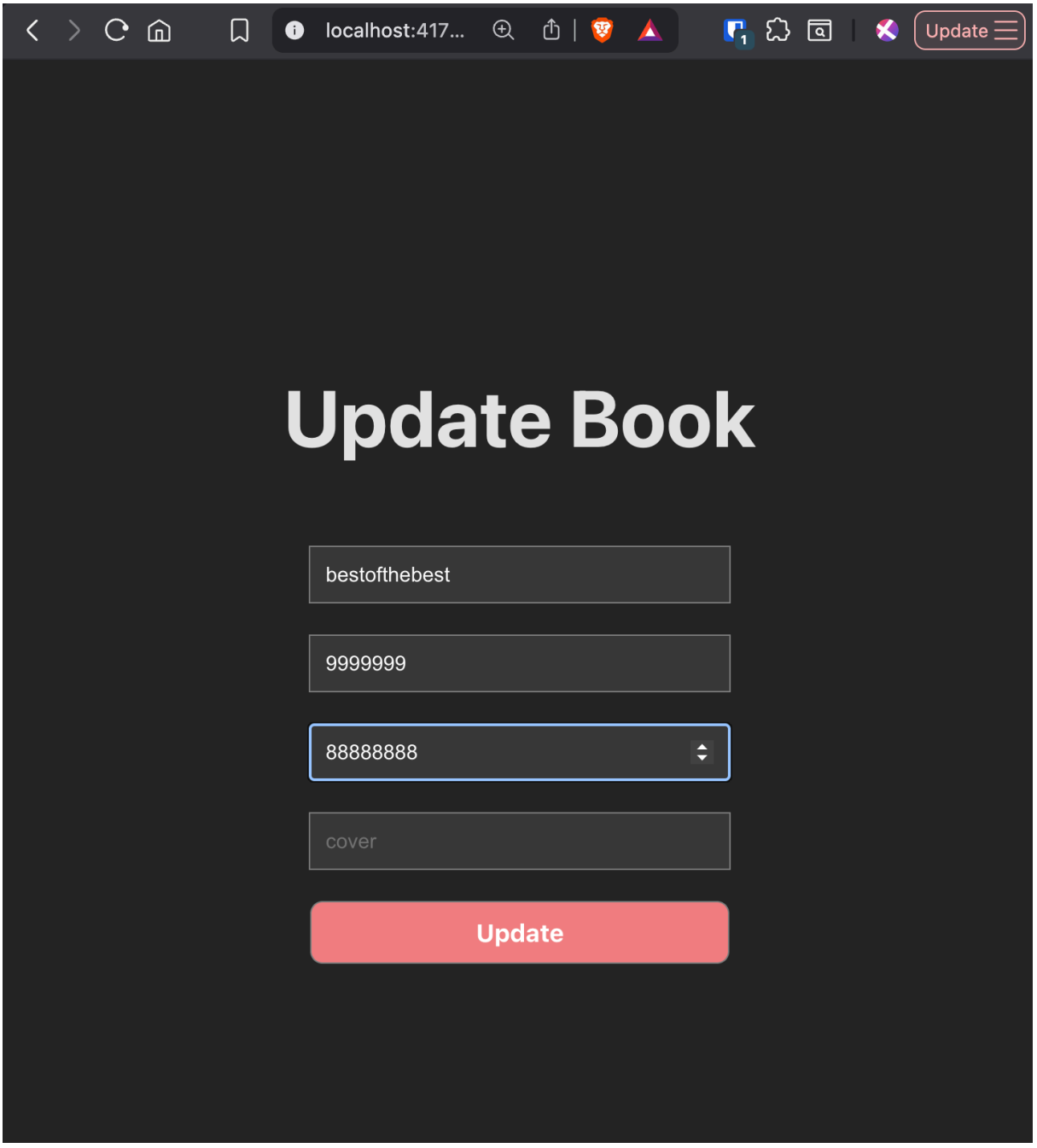


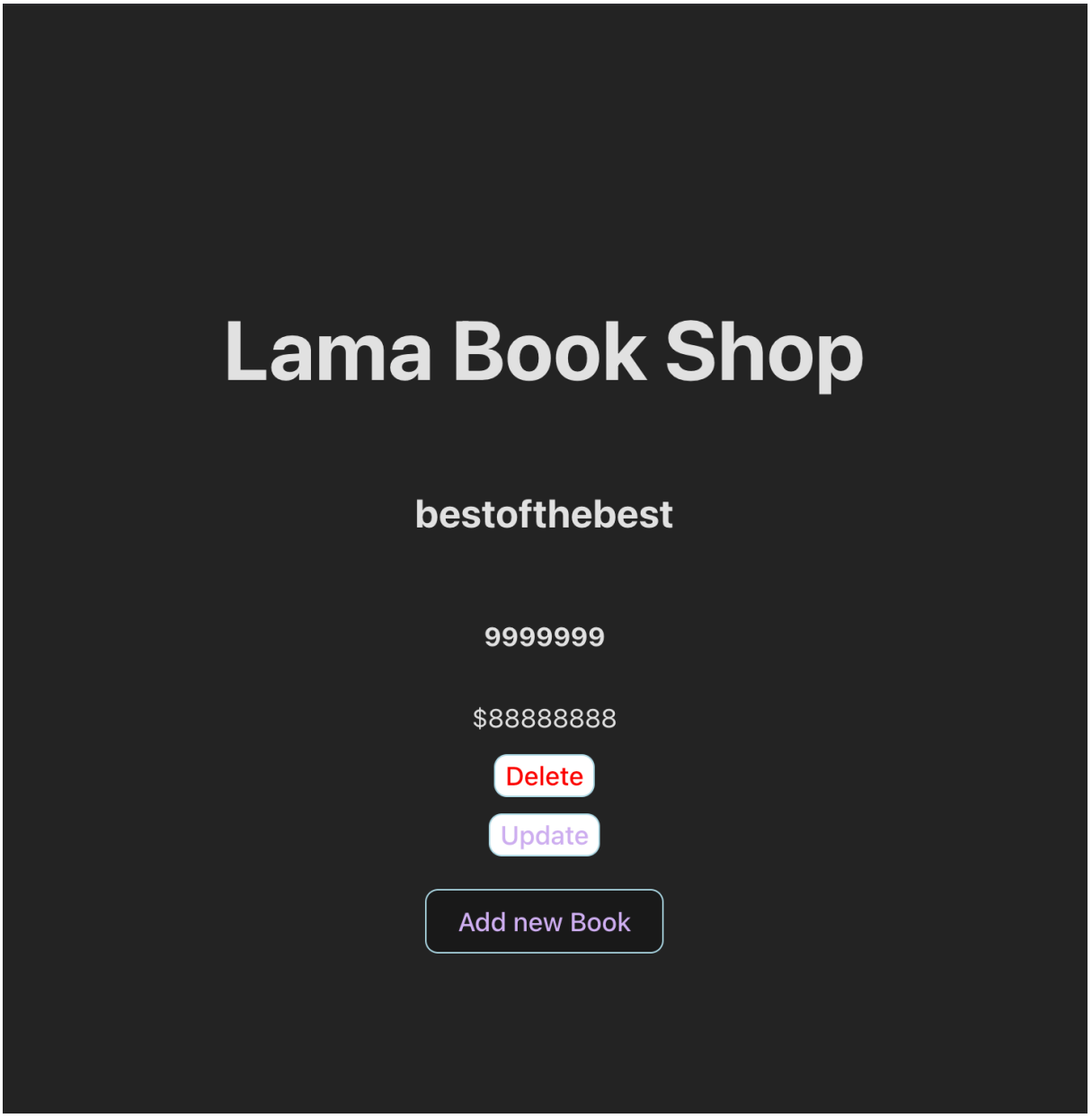
- Create


localhost:417...
Update
Add New Book
title
description
price
cover
Add


localhost:4173
Update
Lama Book Shop
dfasdfds
fdasfds
$111
Delete
Update
Add new Book

- Update


localhost:417...
Update
Update Book
bestofthebest
9999999
88888888
cover
Update


Lama Book Shop
bestofthebest
9999999
$88888888
Delete
Update
Add new Book