**Hausaufgae Dokumentation:**

1) **Frontend erstellen (für produkte)**

**1.1 npx** create-react-app **my-frontend**

**1.2 components/ProductList.js**

import React from 'react';

*function* ProductList() {

*const* [products, setProducts] = React.useState([]);

  React.useEffect(() *=>* {

    fetch('/api/products')

      .then(*response* *=>* *response*.json())

      .then(setProducts);

  }, []);

  return (

    <div>

      <h1>Products</h1>

      {products.map(*product* *=>* (

        <div key={*product*.id}>

          <h2>{*product*.name}</h2>

          <p>{*product*.description}</p>

        </div>

      ))}

    </div>

  );

}

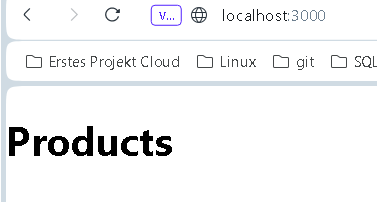
export default ProductList;

**1.3 src/App.js**

import React from 'react';

import ProductList from './components/ProductsList';

*function* App() {

  return (

    <div className="App">

      <*ProductList* />

    </div>

  );

}

export default App;

**2) Create “my-backend” folder**

* **npm** install express
* **npm** init -y
* **npm** install cors
* **npm** install mysql2

*const* express = require('express');

*const* mysql = require('mysql2/promise');

*const* cors = require('cors');

*const* app = express();

app.use(cors()); // Enable CORS

app.use(express.json());

*let* pool = mysql.createPool({

    host: 'localhost',

    user: 'root',

    password: '',

    database: 'my\_database',

    waitForConnections: true,

    connectionLimit: 10,

    queueLimit: 0

});

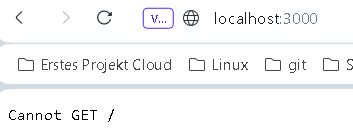
app.get('/api/products', async (*req*, *res*) *=>* {

*const* [rows, fields] = await pool.execute('SELECT \* FROM products');

*res*.json(rows);

});

app.listen(3001, () *=>* console.log('Server ist auf Port 3001'));

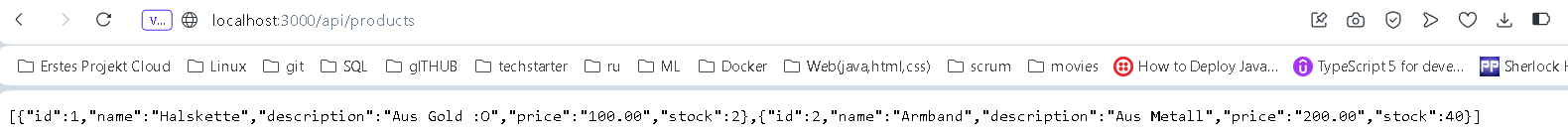


Gut, in der Zukunft zu testen mit API-Tools wie z.b mit **Postman**

3) **Datenbank einrichten MIT MySQL $$$**

|  |  |  |
| --- | --- | --- |
|  | CREATE TABLE products (    id *INT* AUTO\_INCREMENT PRIMARY KEY,    name *VARCHAR*(255),    description *TEXT*,    stock *INT*  ); | „Spalten erstellen“ mit  id, name, description & stock |
| Was eingefügt…. | | |

**Testen…..**

* **node** server.js

**Cool, backend funktioniert auch hier^^**

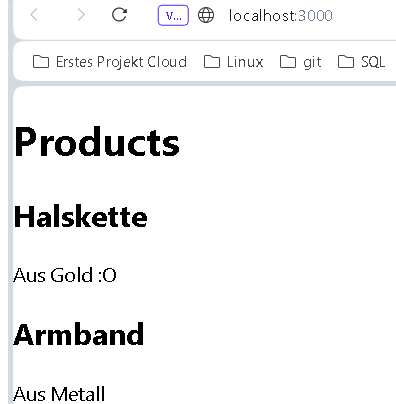
**Im server.js ergänzen**

* npm install cors 🡪 um mit frontend zu kommunizieren

**Im Frontend**: **npm** start

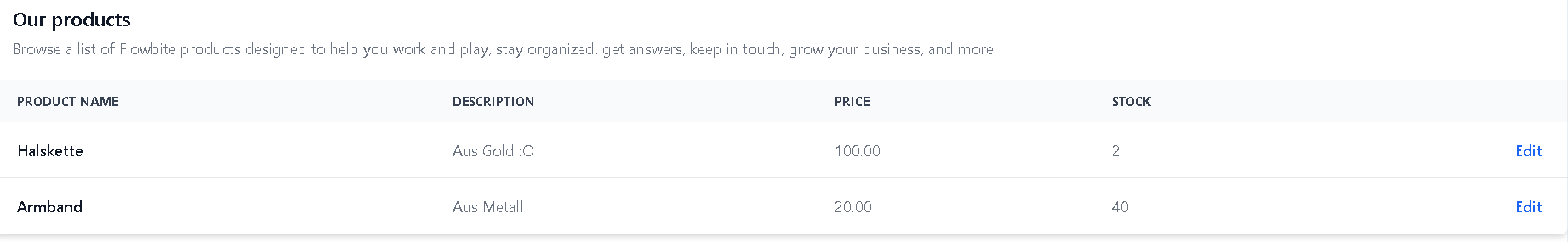
**Backend:** **node** server.js

* Beachte Backend muss mit node.js & frontend muss mit npm start **aktiv** sein

**Frontend Funktioniert auch Hier^^**

Bisschen Verschönern (TailwiND)

Nun ist es Zeit die **Anwendung zu Containereresieren**



4) Da alles lokal funktioniert ist es an der Zeit die Anwendung in **Container zu verpacken,** dazu hab ich eine docker compose mit Dockerfiles vorbereitet damit es alles im **selben netzwerk** Startet

Dockercompose weil ich mit phpadmin Zugang zur Datenbank brauche mit **weniger aufwand**..

|  |  |
| --- | --- |
| Frontend Dockerfile | Backend Dockerfile |
| FROM node:14  WORKDIR /app  COPY package\*.json ./  RUN npm install  COPY . .  EXPOSE 3000  CMD ["npm", "start"] | FROM node:14  WORKDIR /app  COPY package\*.json ./  RUN npm install  COPY . .  EXPOSE 3001  CMD ["node", "server.js"] |
| version: '3'  services:  my-frontend:  build: ./my-frontend  ports:  - "3000:3000"  my-backend:  build: ./my-backend  ports:  - "3001:3001"  my-database:  image: mysql:5.7  environment:  - MYSQL\_ROOT\_PASSWORD=rootpassword  - MYSQL\_DATABASE=mydatabase  - MYSQL\_USER=username  - MYSQL\_PASSWORD=password  phpmyadmin:  image: phpmyadmin/phpmyadmin  ports:  - '8080:80'  environment:  PMA\_HOST: my-database  PMA\_PORT: 3306 | |

Passe an…

*const* pool = mysql.createPool({

    host: 'my-database', // changee

    user: 'username',

    password: 'password',

    database: 'my\_database', // change

    waitForConnections: true,

    connectionLimit: 10,

    queueLimit: 0

  });

Damit es im Containernetzwerk kommunizieren kann

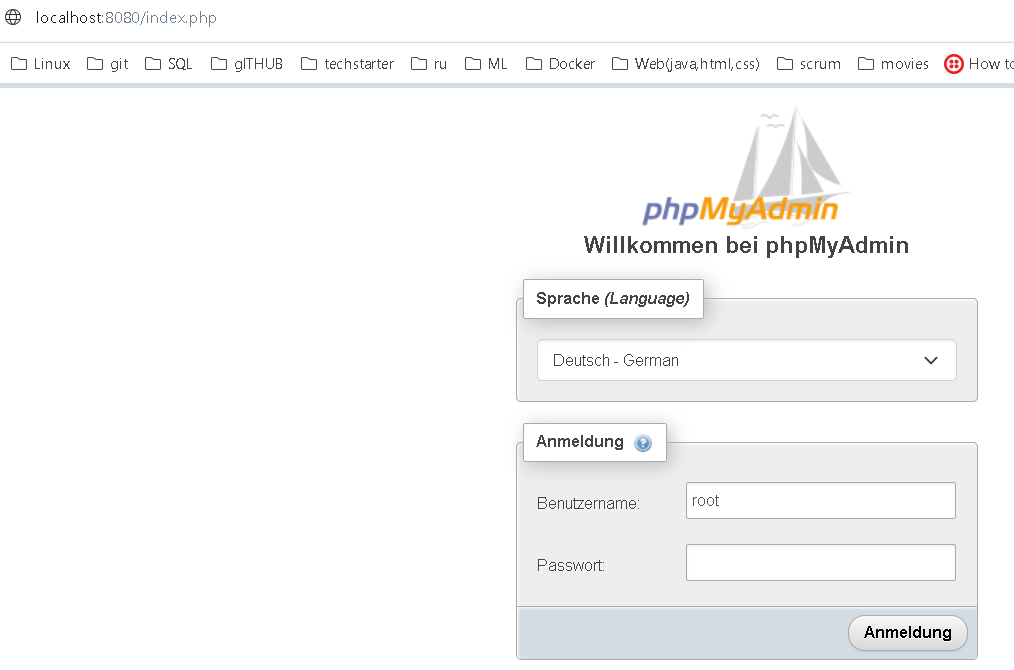
Nun umgebungsvariablen anpassen

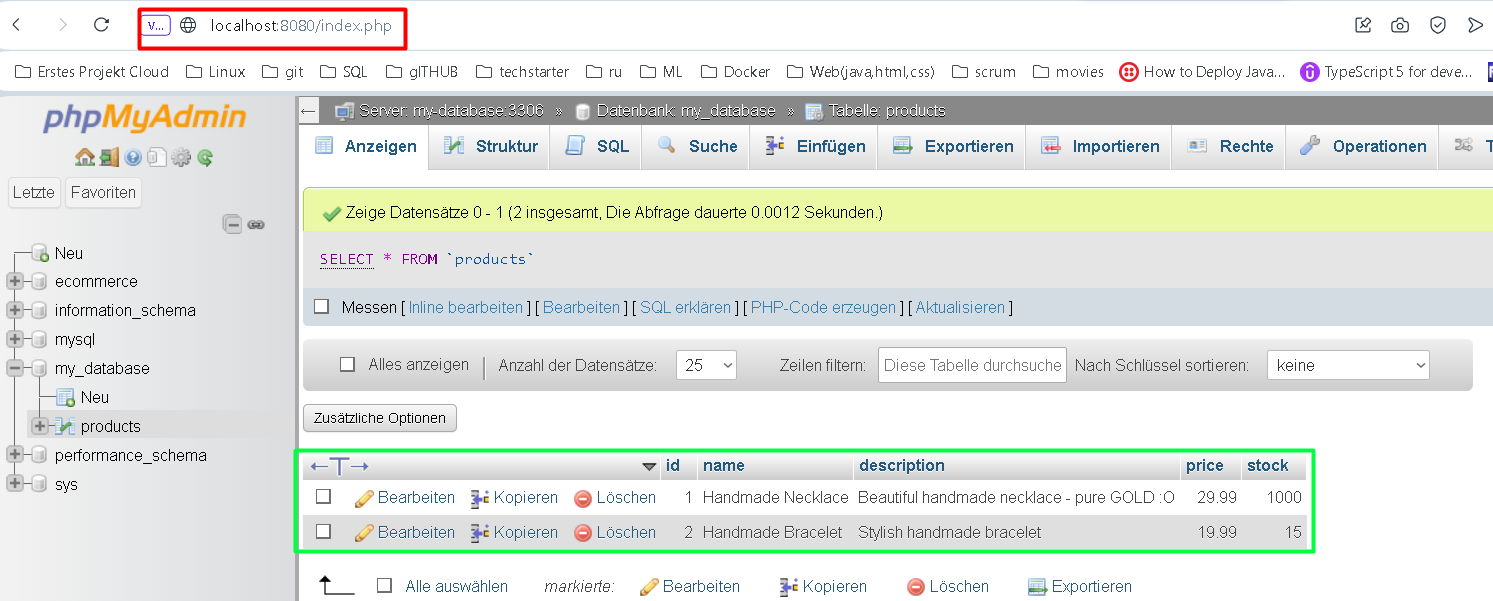
|  |  |  |
| --- | --- | --- |
|  | Server.js | ProduktList.js |
| davor | *const* db = mysql.createConnection({  //    host: 'my-database',    user: 'username',    password: 'password',    database: 'my\_database'  }); | useEffect(() *=>* {  fetch('http://localhost:3001/api/products')  .then(*response* *=>* *response*.json())  .then(setProducts)  .catch(*error* *=>* console.error('Error:', *error*));  }, []); |
| Im env. | MYSQL\_HOST=my-database  MYSQL\_USER=username  MYSQL\_PASSWORD=password  MYSQL\_DATABASE=my\_database | ACT\_APP\_BACKEND\_URL=http://localhost:3001 |
|  |  |  |

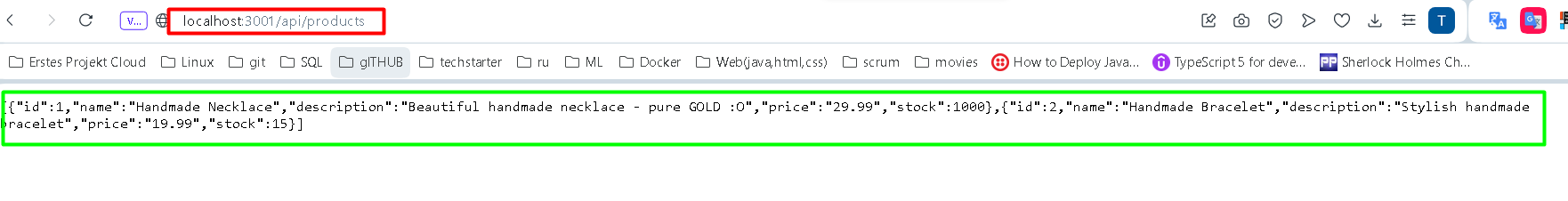
Nun docker-compose.yaml --build

* Build damit es nacheinander build

**MySql-Datenbasis**





**2) Mein Backend**

3) Frontend

