

Laporan Progres Mingguan

Kelompok: 10

Mitra: Jaya Mandiri Motor

Pekan ke-: 13

Tanggal: 9 Mei 2025

Anggota:

- Dzaky Rasyiq Zuhair_10231035_QA & DevOps
- Muhammad Khoiruddin Marzuq_10231065_Frontend Developer
- Riska Fadlun Khairiyah Purba_10231083_Project Manager & UI/UX Designer
- Tiya Mitra Ayu Purwanti_10231088_Backend Developer

Progress Summary

Pada pekan ke-13, tim berhasil menambahkan fitur baru ke dalam sistem, yaitu fitur admin dan order. Tim berhasil membuat fitur admin yang dapat memanajemen penggunaan dan pengaturan sistem, sementara fitur orders digunakan untuk menangani alur pemesanan. Penambahan ini dapat meningkatkan dan memperkuat pengembangan fitur selanjutnya.

Accomplished Tasks

Core Feature #4 - Fitur Utama orders

- Fitur orders berguna untuk menangani alur terkait pemesanan, seperti membuat pesanan baru, melihat daftar pesanan, dan memantau status pesanan.

Admin panel

- Fitur admin dalam sistem berguna untuk manajemen data, Admin dapat melihat, mengubah, atau menghapus data seperti, produk, pesanan, kategori dan pengguna.
- Fitur admin berfungsi untuk membatasi akses ke fungsi-fungsi tertentu agar hanya dapat diakses oleh user dengan peran admin.

Frontend Development

- Implementasi tampilan CRUD Admin dan fitur order
- Form create/update produk terhubung ke backend
- Responsivitas tampilan di berbagai ukuran layar

Backend Development

- Pembuatan tabel Ordes
- Endpoint CRUD admin dan fitur order

Challenges & Solutions

Tantangan: Terjadinya kesalahan pada file package-lock.json, sehingga terjadi eror **Solusi:** Solusi yang kami lakukan adalah menghapus file package-lock.json beserta folder node_modules, Kemudian menjalankan ulang perintah npm instal, dengan tujuan untuk memberishkan dari maslaha dan membuat ulang package-lock.json secara otomatis berdasarkan daftar paket yang ada di package-lock.json.

Next Week Plan

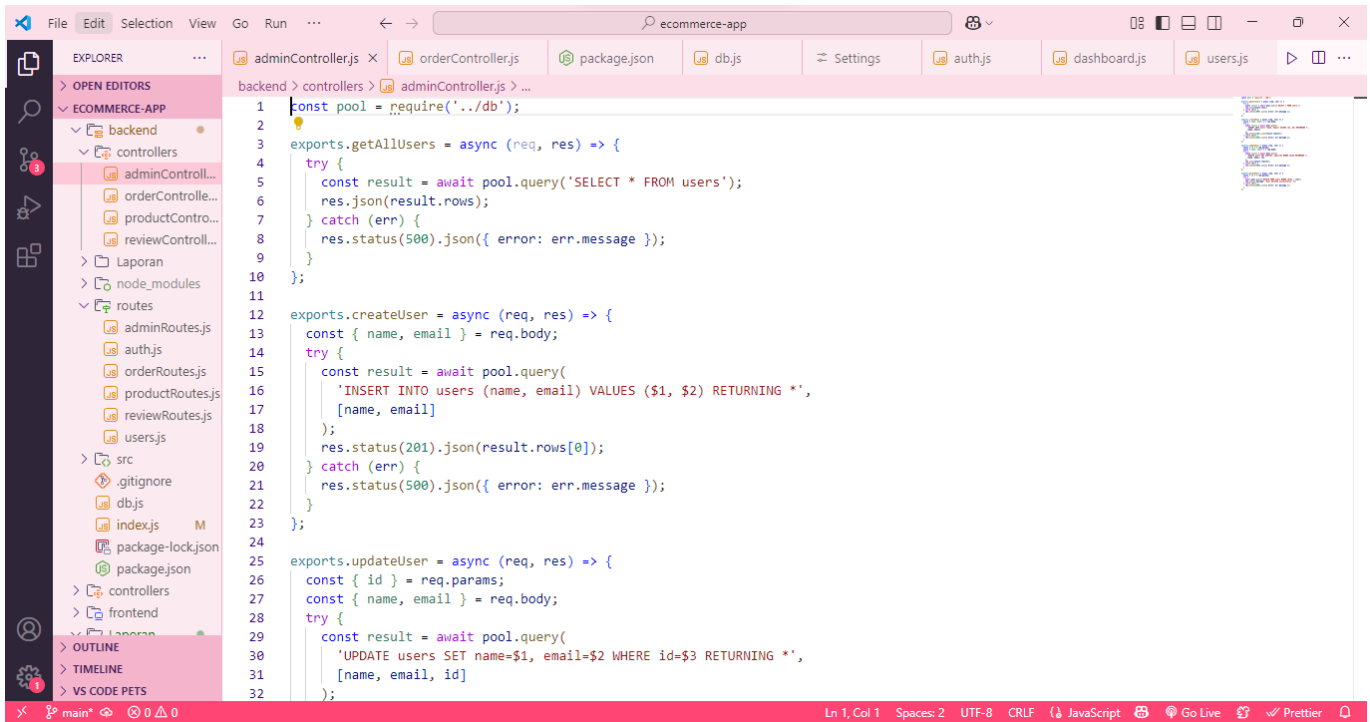
- Menambahkan fitur sesuai rencana awal yang sudah di susun dengan kesepakatan mitra.
 - Memasukin tahap penyempurnaan seluruh fitur
 - Bugfixing Usability testing Persiapan deployment (jika diperlukan) Demo progress ke mitra
-

Contributions

- **Riska Fadlun Khairiyah P (Project Manager & UI/UX Designer):** Menyusun dokumentasi, mengkoordinasi alur kerja CRUD produk dan review, memastikan konsistensi antara frontend dan backend, Membantu membackeup pengembangan CRUD admin dan fitur orders.
 - **Dzaky Rasyiq Zuhair (QA & DevOps):** Melakukan testing pada operasi CRUD, debugging pada proses update dan delete produk, menyiapkan environment testing untuk semua endpoint
 - **Muhammad Khoiruddin Marzuq (Frontend Developer):** Memastikan tampilan responsive untuk semua operasi CRUD, mengimplementasikan tampilan untuk operasi produk, membuat form untuk create/update produk
 - **Tiya Mitra Ayu Purwanti (Backend Developer):** Membuat struktur tabel products dan reviews, mengembangkan semua endpoint API, menangani operasi delete produk, mengimplementasikan logika update stok produk, membangun koneksi database.Membantu menyusuk dokumentasi dan koorniasi tim.
-

Screenshots/Demo

a. Menambahkan kode baru



```

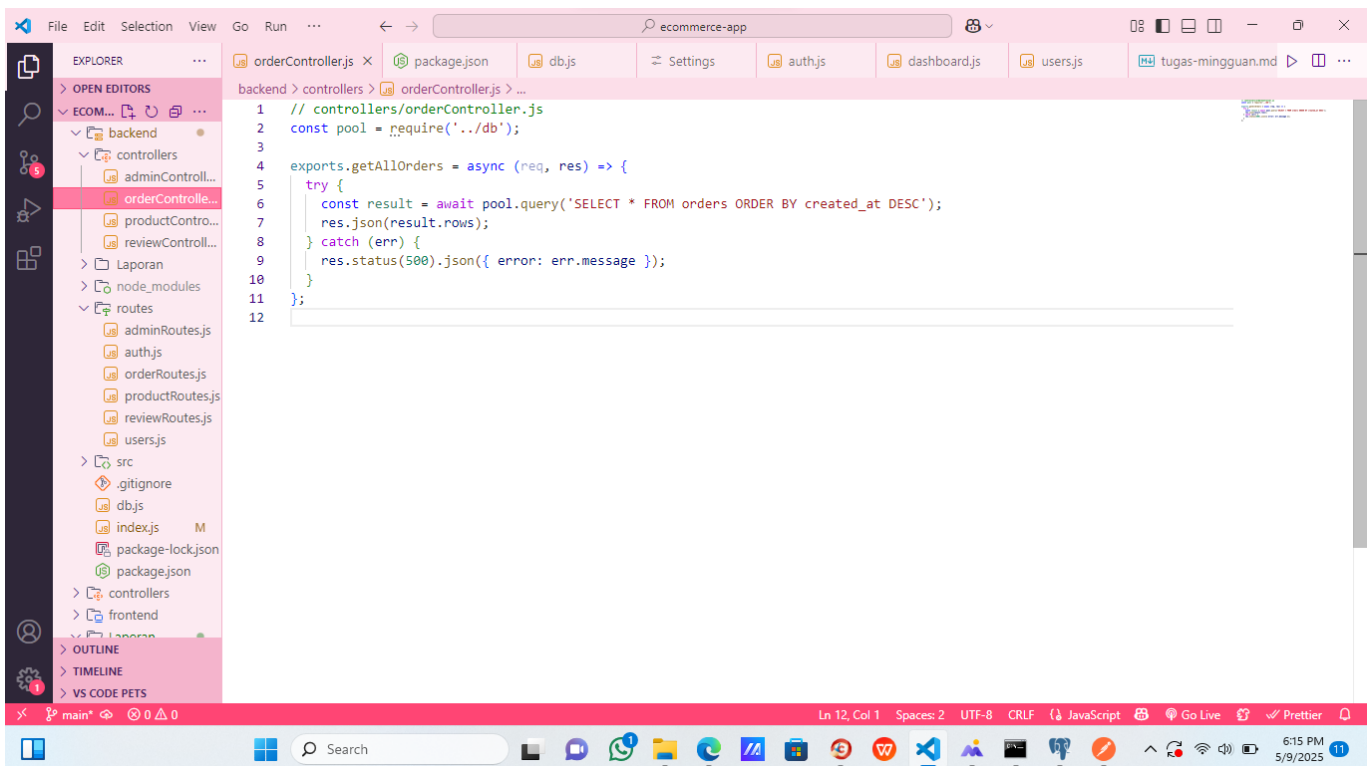
1  const pool = require('../db');
2
3  exports.getAllUsers = async (req, res) => {
4    try {
5      const result = await pool.query('SELECT * FROM users');
6      res.json(result.rows);
7    } catch (err) {
8      res.status(500).json({ error: err.message });
9    }
10   };
11
12   exports.createUser = async (req, res) => {
13     const { name, email } = req.body;
14     try {
15       const result = await pool.query(
16         'INSERT INTO users (name, email) VALUES ($1, $2) RETURNING *',
17         [name, email]
18       );
19       res.status(201).json(result.rows[0]);
20     } catch (err) {
21       res.status(500).json({ error: err.message });
22     }
23   };
24
25   exports.updateUser = async (req, res) => {
26     const { id } = req.params;
27     const { name, email } = req.body;
28     try {
29       const result = await pool.query(
30         'UPDATE users SET name=$1, email=$2 WHERE id=$3 RETURNING *',
31         [name, email, id]
32       );
33     } catch (err) {
34       res.status(500).json({ error: err.message });
35     }
36   };

```

Gambar di atas adalah berisi kode yang digunakan untuk menangani fungsi-fungsi yang hanya dimiliki oleh admin, fungsi tersebut berguna untuk mengambil, membuat, memperbarui, dan menghapus data dari user.

 menambahkan kode adminRoutes.js

Gambar di atas adalah berfungsi untuk menangkap permintaan atau menangani permintaan terkait admin dengan menghubungkan ke Controller yang akan diakses oleh tim frontend.

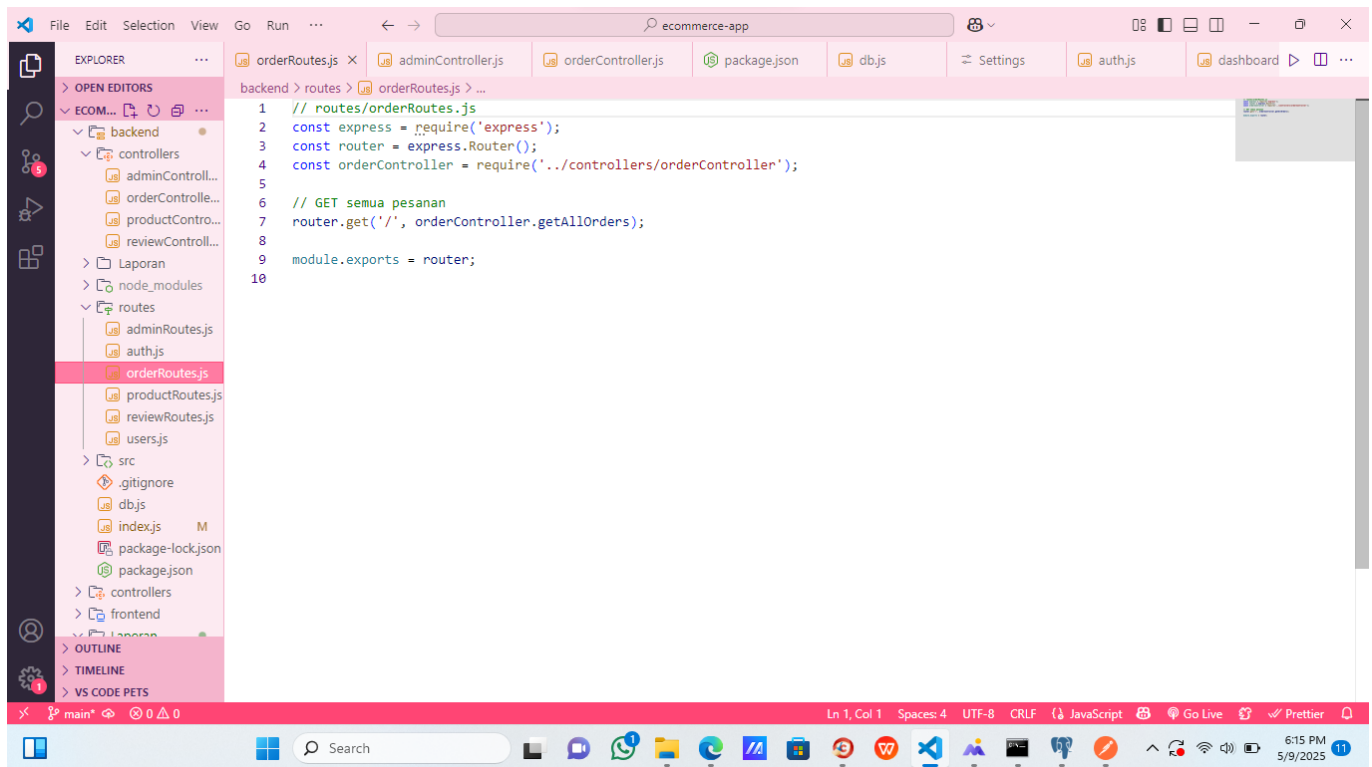


```

1  // controllers/orderController.js
2  const pool = require('../db');
3
4  exports.getAllOrders = async (req, res) => {
5    try {
6      const result = await pool.query('SELECT * FROM orders ORDER BY created_at DESC');
7      res.json(result.rows);
8    } catch (err) {
9      res.status(500).json({ error: err.message });
10   }
11 }
12

```

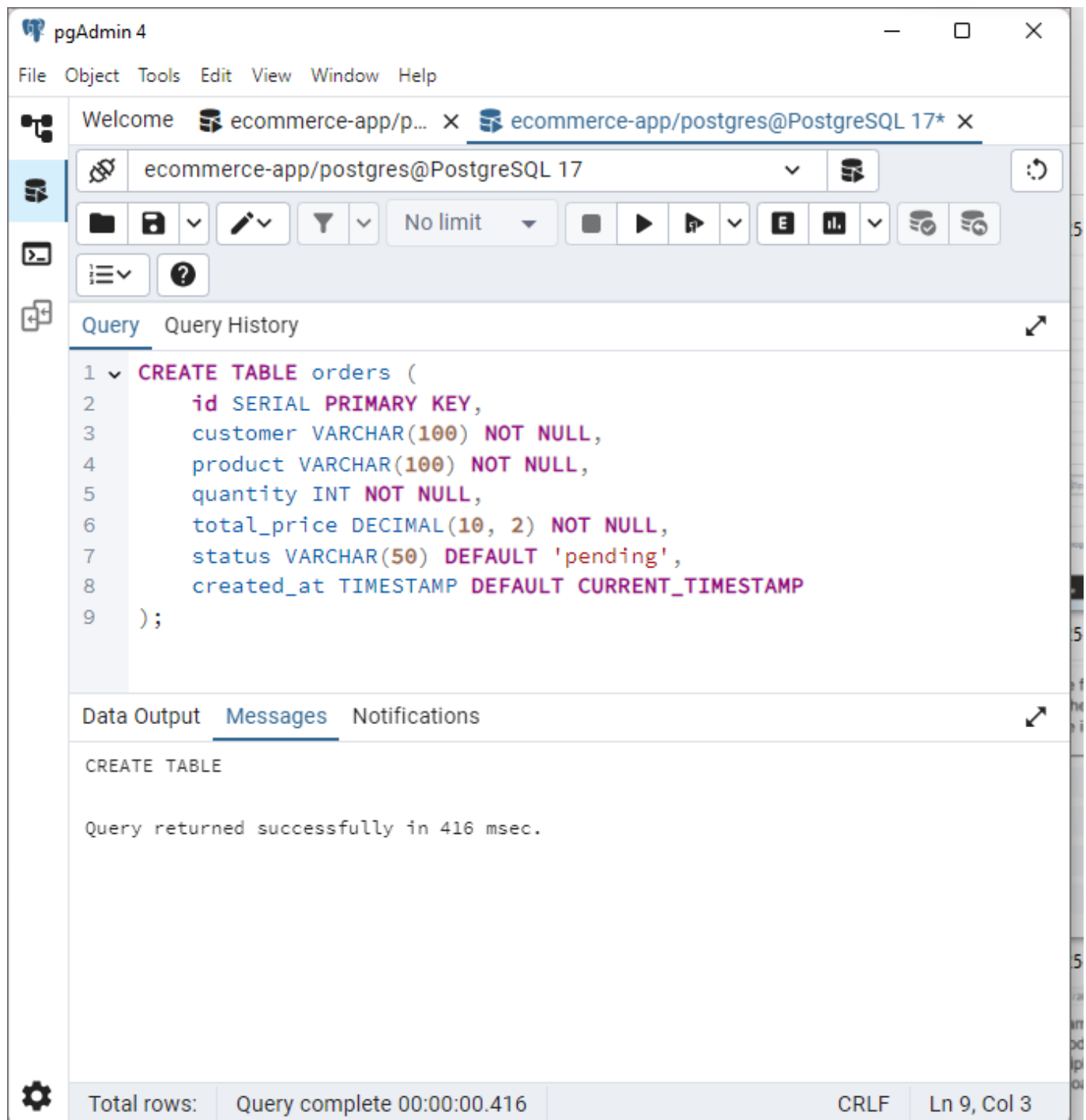
Gambar di atas adalah berfungsi untuk menangani entitas Order dalam sebuah web yang memiliki layanan pemesanan.



Gambar di atas adalah berfungsi untuk menghubungkan ke controller.

b. Pembuatan Tabel database

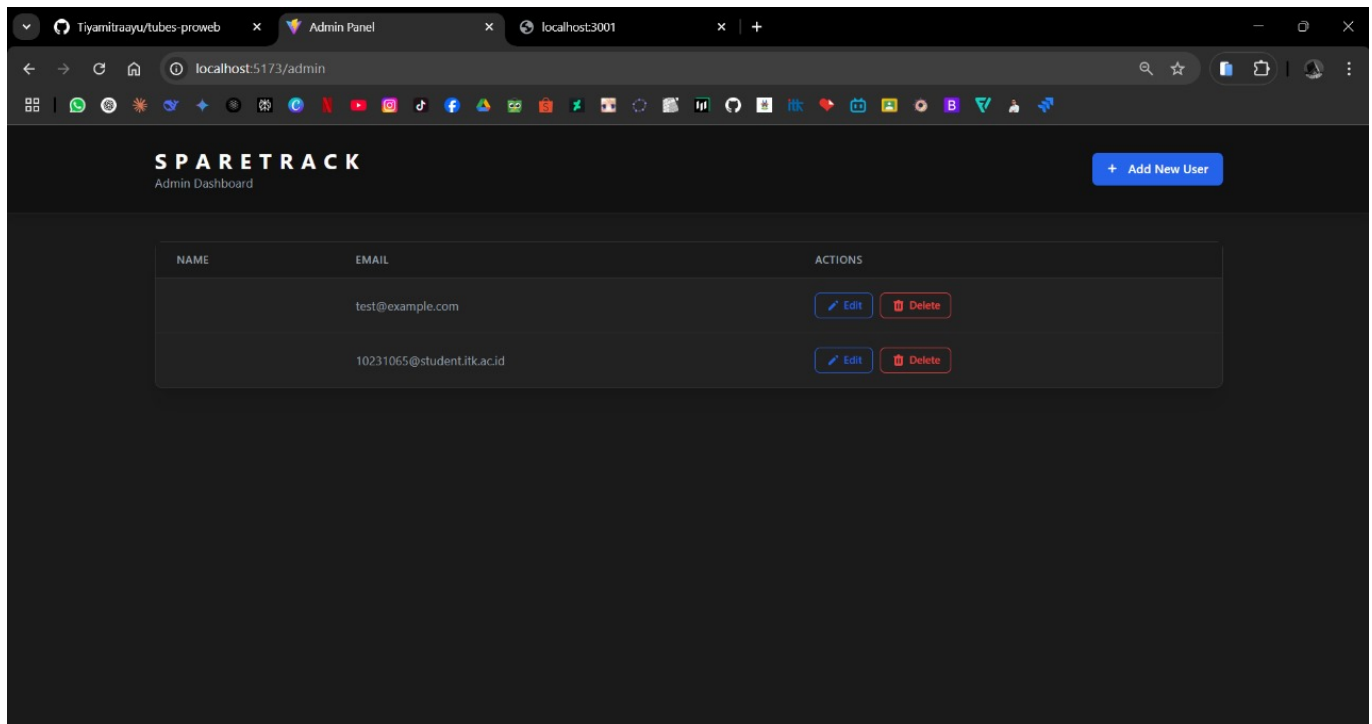
Gambar di bawah menunjukkan tahap pembuatan tabel produk dan review di PostgreSQL.

Tabel database orders:

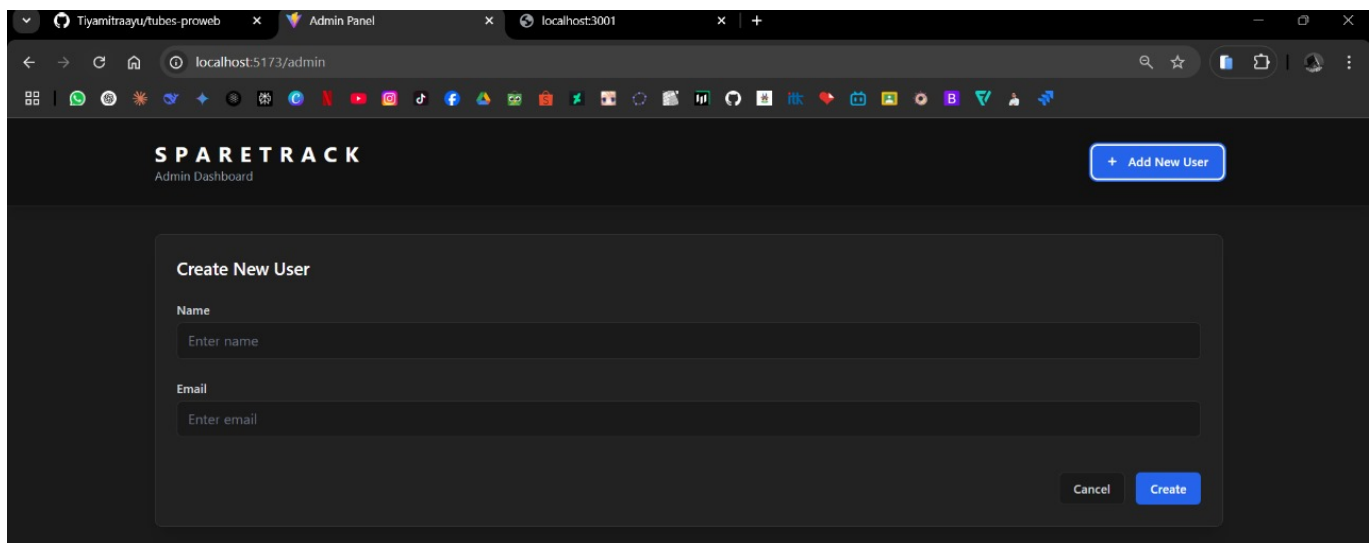
Struktur tabel **products** dengan kolom:

- id
- customer
- product
- quantity
- total_price
- status
- created_at

b. Tampilan CRUD admin



Gambar diatas menunjukkan tampilan dari fungsi baru yang sudah berhasil kami kembangkan, tampilan di atas adalah fitur yang hanya dapat di akses oleh user dengan peran admin. Pada halaman yang ditampilkan diatas memiliki fungsi yang berbeda beda, tampilan diatas sudah di lengkapi stayling visual.



Gambar diatas lampiran tambahan.