

# Laporan Progres Mingguan - **SmartBizAdmin**

---

## Kelompok: 5

- **Muhammad Fikri Haikal Ariadma / 10231063**
- **Irfan Zaki Riyanto / 10230145**
- **Micka Mayulia Utama / 10231053**
- Ika Agustin Wulandari / 10231041\*\*

 Mitra: Kost Al-Fitri D'Carjoe

 Pekan ke-: 10

 Tanggal: 18/04/2025

---

## Progress Summary

Pada pekan ke-10, tim **SmartBizAdmin** telah menyelesaikan perancangan dan implementasi awal backend untuk sistem Smartbiz Admin, termasuk desain dan setup database PostgreSQL serta pembuatan REST API skeleton. Struktur frontend dasar juga telah disiapkan.

---

## Accomplished Tasks

-  Merancang skema database Smartbiz Admin
  -  Mengimplementasikan struktur database di PostgreSQL
  -  Menyiapkan struktur folder backend
  -  Menambahkan koneksi database melalui file `db.js`
  -  Membuat REST API skeleton dengan Express.js (endpoint dasar untuk login, register, keuangan, kos, coffee shop, dan inventaris)
  -  Menyiapkan struktur frontend dasar dengan React + Vite
- 

## Challenges & Solutions

-  **Challenge 1:** Penyesuaian struktur folder dan routing backend  
 **Solution:** Menyesuaikan struktur folder agar modular dan sesuai standar Express, serta membuat `index.js` yang mengimpor dan menjalankan semua route secara konsisten.
  -  **Challenge 2:** Koneksi ke database PostgreSQL  
 **Solution:** Membuat file `db.js` yang menggunakan modul `pg` untuk membuat koneksi pool yang dapat digunakan oleh seluruh route.
- 

## Next Week Plan

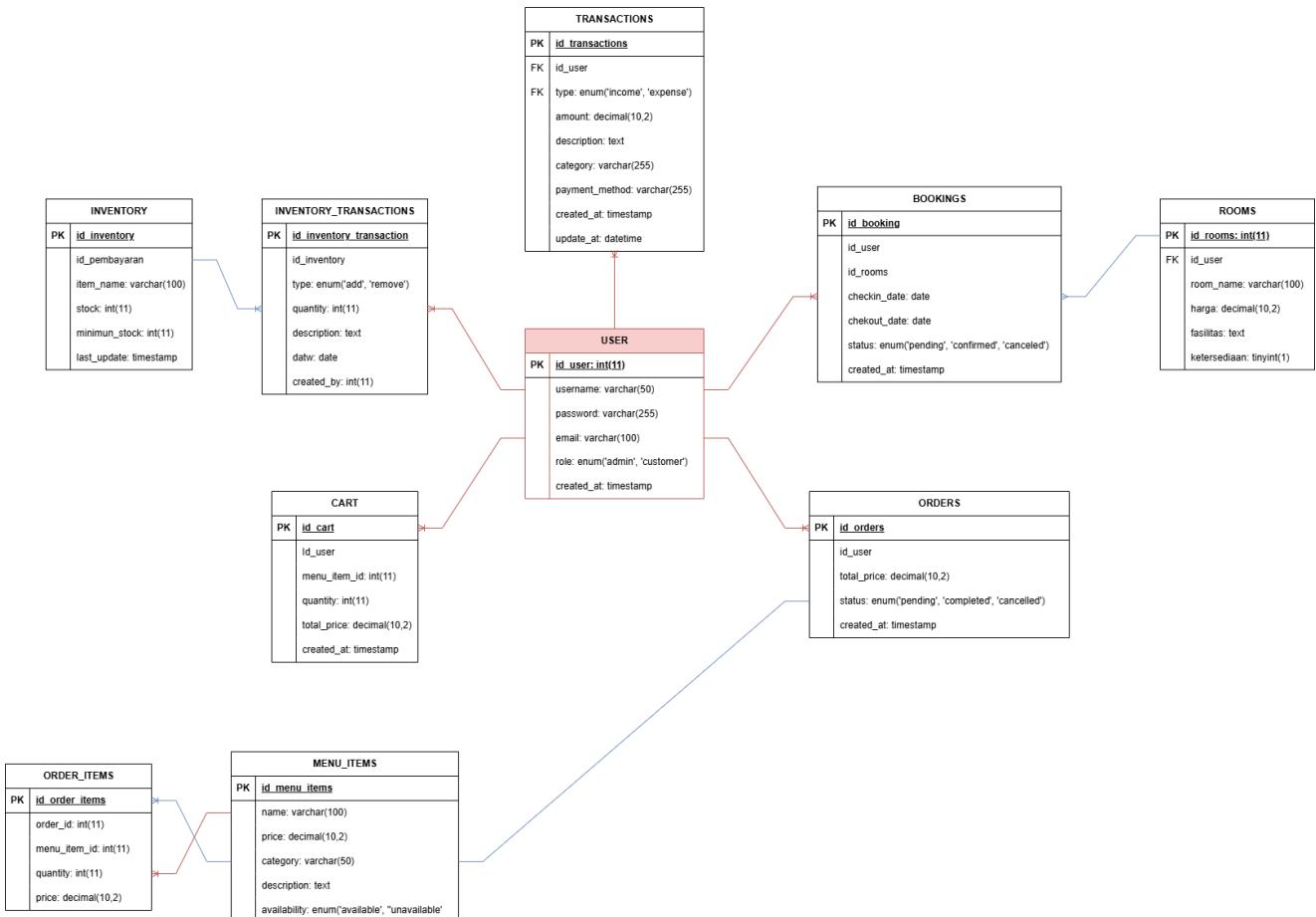
- Implementasi sistem autentikasi (login/register)
  - Implementasi fitur inti #1 (sesuai kebutuhan mitra)
  - Integrasi frontend-backend untuk fitur yang sudah ada
  - Demo progress ke mitra
- 

## Contributions

- **Muhammad Fikri Haikal Ariadma / 10231063**  
→ Membuat struktur folder frontend, merancang skema database, mengimplementasikan struktur database
  - **Irfan Zaki Riyanto / 10230145**  
→ Menyusun struktur backend, membuat koneksi database, menambahkan routing dasar, merancang skema database, menyusun laporan md
  - **Micka Mayulia Utama / 10231053**  
→ Merancang skema database, menyusun laporan md
  - **Ika Agustin Wulandari / 10231041**  
→ Merancang skema database
- 

## Screenshots / Demo

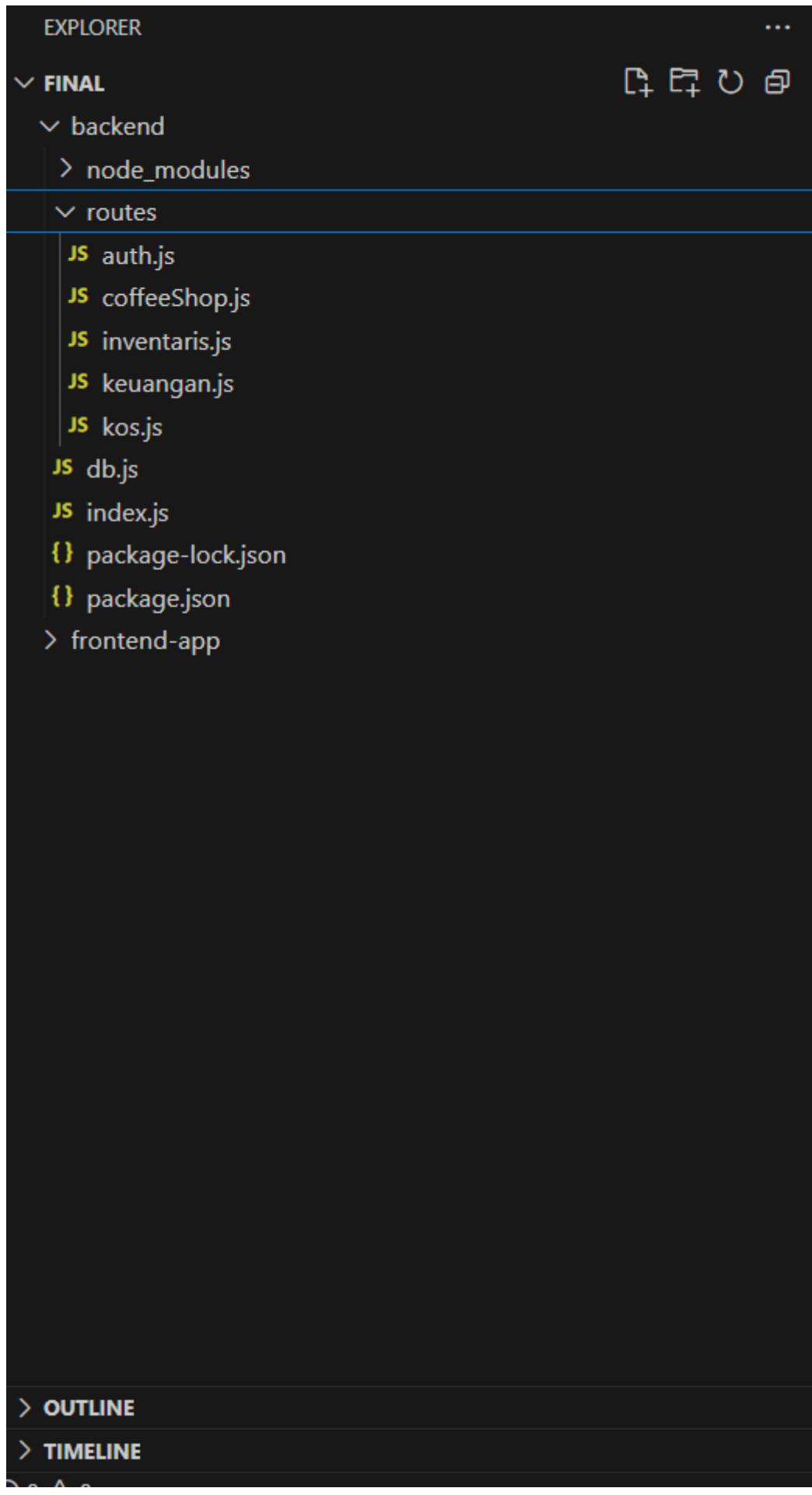
### Skema Database:



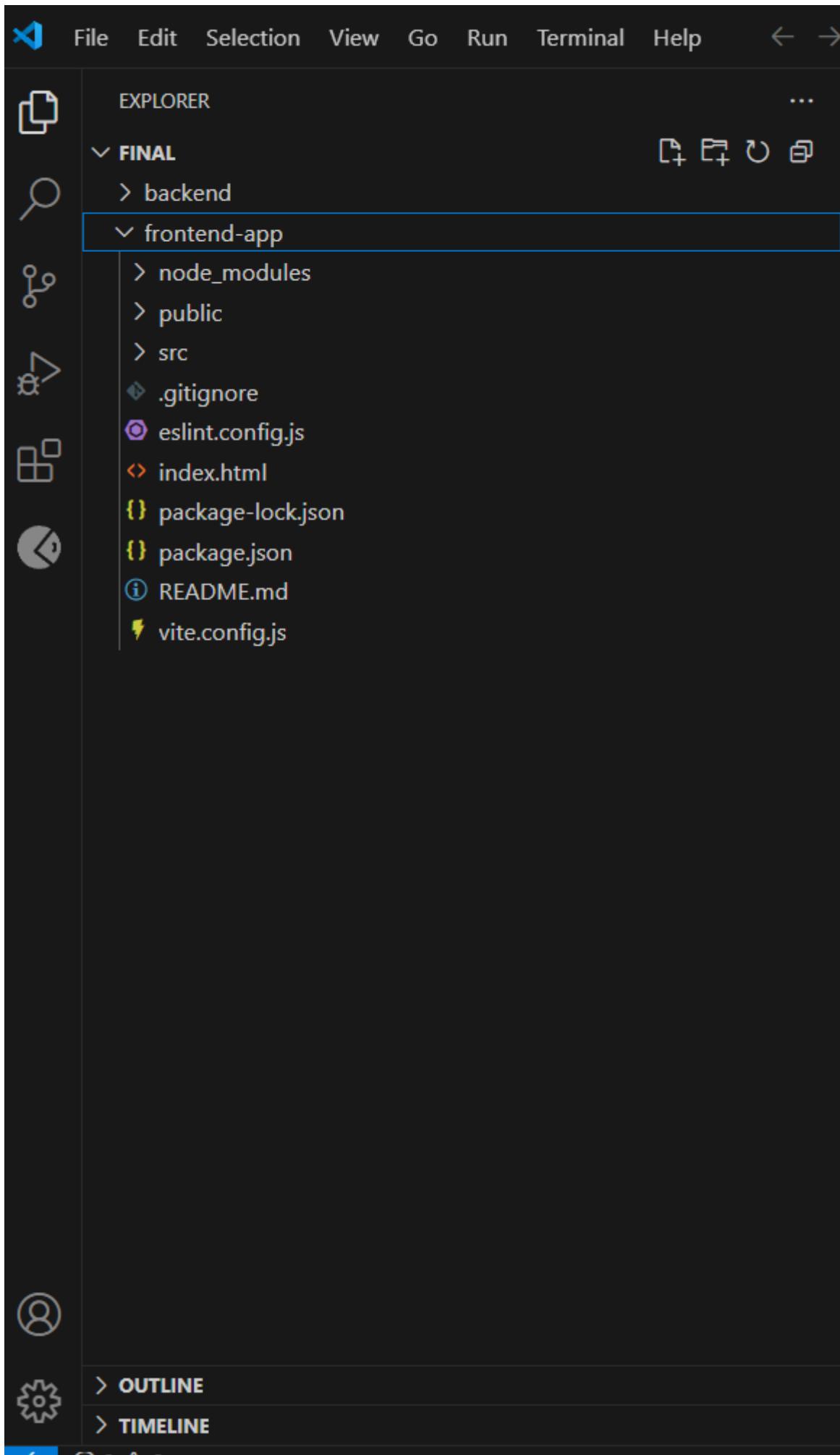
Gambar di atas menunjukkan skema database **SmartBizAdmin**, yang terdiri dari 10 tabel utama:

- **Tabel user**
  - Memiliki relasi 1:N ke **cart**, **orders**, **booking**, **transaction**, **inventory transaction**, dan **order items**
- **Tabel cart**
- **Tabel menu\_items**
  - Memiliki relasi 1:N ke **order items**
- **Tabel order items**
- **Tabel orders**
- **Tabel booking**
- **Tabel rooms**
  - Memiliki relasi 1:N ke **booking**
- **Tabel transaction**
- **Tabel inventory**
- **Tabel inventory transaction**

## 💡 Struktur Backend:



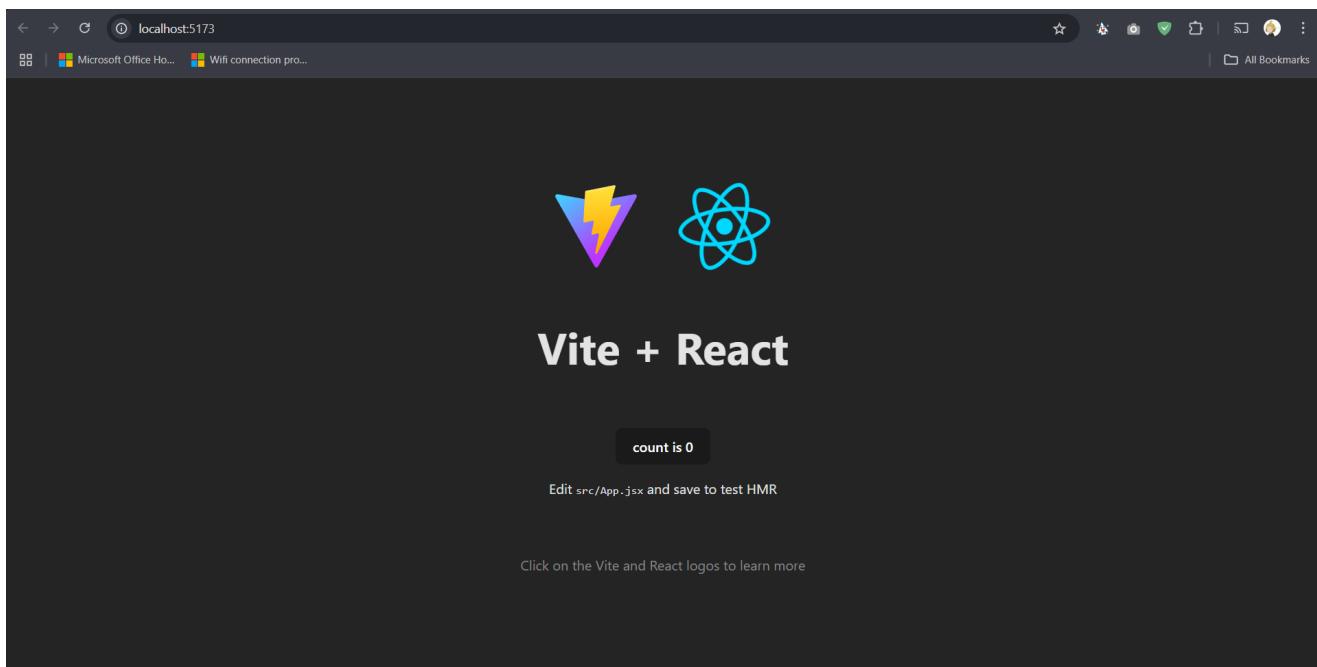
## Struktur Frontend:



Struktur frontend dasar menggunakan React + Vite.

---

### ⌚ Tampilan Frontend:



Halaman awal dari sistem **SmartBizAdmin** yang berhasil dirender menggunakan React + Vite.

---

### ⌚ Koneksi Database (db.js):

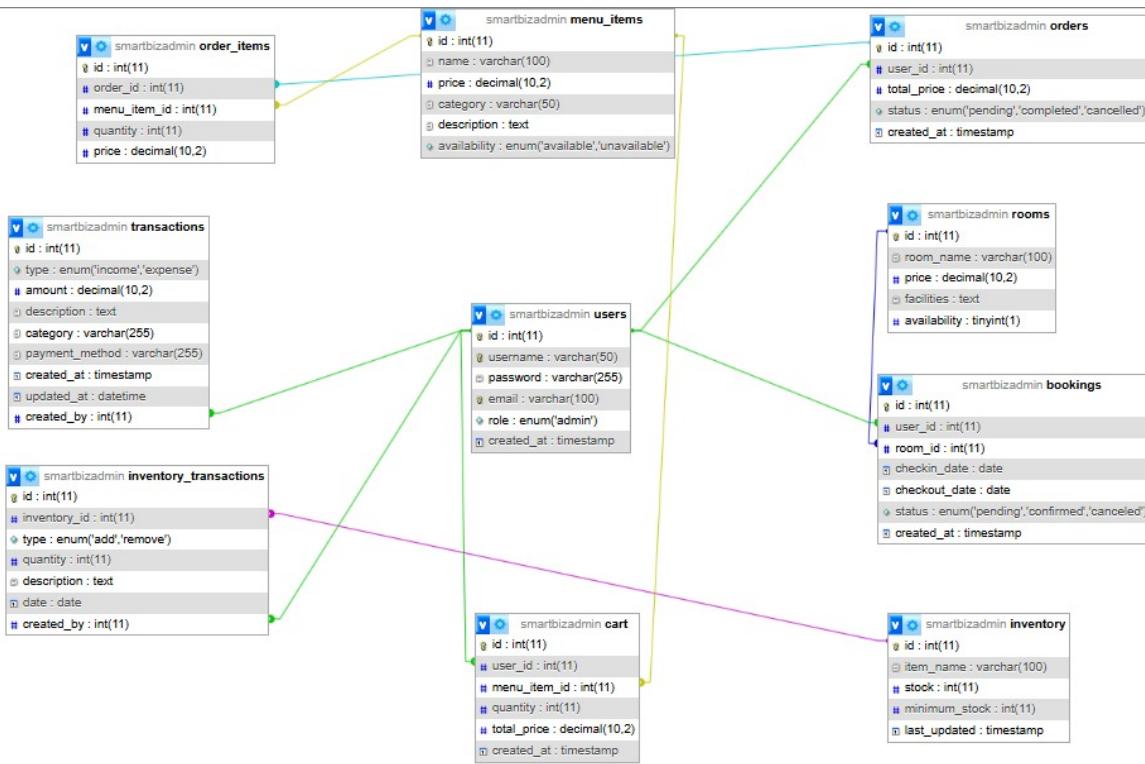
A screenshot of a code editor showing a file named "db.js". The code is written in JavaScript and uses the pg module to connect to a PostgreSQL database. The connection pool is configured with various parameters like user, host, database, password, and port. The code is as follows:

```
backend > JS db.js > ...
1 // db.js
2 const { Pool } = require('pg');
3
4 const pool = new Pool({
5   user: 'postgres',          // Sesuaikan jika username berbeda
6   host: 'localhost',
7   database: 'Smartbiz-Admin', // Nama database kamu
8   password: 'postgres',      // Ganti jika password PostgreSQL kamu berbeda
9   port: 5432,
10 });
11
12 module.exports = pool;
13
```

db.js digunakan untuk mengatur koneksi backend ke PostgreSQL menggunakan modul pg dengan pool yang bisa digunakan oleh semua route.

---

### ⌚ Implementasi Struktur Database (ERD):



Gambar ini menunjukkan hubungan antar tabel utama seperti **users**, **transactions**, **inventory**, **menu\_items**, **orders**, **bookings**, hingga **rooms**.

## 💡 REST API Skeleton:

- **Endpoint Coffee Shop:**

```
JS coffeeShop.js X
backend > routes > JS coffeeShop.js > ...
1 const express = require('express');
2 const router = express.Router();
3 const pool = require('../db');
4
5 router.get('/menu', (req, res) => {
6   res.send(`Daftar menu coffee shop berhasil diambil`);
7 });
8
9 router.post('/order', (req, res) => {
10  const { itemName } = req.body;
11  res.send(`Pesanan ${itemName} berhasil ditambahkan`);
12 });
13
14 module.exports = router;
```

Difungsikan untuk Client dalam mengirim request ke server lewat endpoint API. kemudian Server memproses permintaan (bisa ambil data dari database, proses logic, dll). Dan terakhir Server mengirim response kembali ke client

- **Endpoint Inventaris:**

```
JS inventaris.js x
backend > routes > JS inventaris.js > ...
1 const express = require('express');
2 const router = express.Router();
3 const pool = require('../db');
4
5 router.get('/', (req, res) => {
6   res.send('Data inventaris berhasil diambil');
7 });
8
9 router.post('/', (req, res) => {
10   const { itemName } = req.body;
11   res.send(`Item ${itemName} berhasil ditambahkan ke inventaris`);
12 });
13
14 module.exports = router;
```

API yang digunakan untuk mengelola data inventaris barang pada suatu sistem kost dan juga coffee shop

- **Endpoint Kos:**

```
JS kos.js x
backend > routes > JS kos.js > ...
1 const express = require('express');
2 const router = express.Router();
3 const pool = require('../db');
4
5 router.get('/list', (req, res) => {
6   res.send('Daftar kamar kos berhasil diambil');
7 });
8
9 router.post('/add', (req, res) => {
10   const { roomName } = req.body;
11   res.send(`Kamar ${roomName} berhasil ditambahkan`);
12 });
13
14 module.exports = router;
```

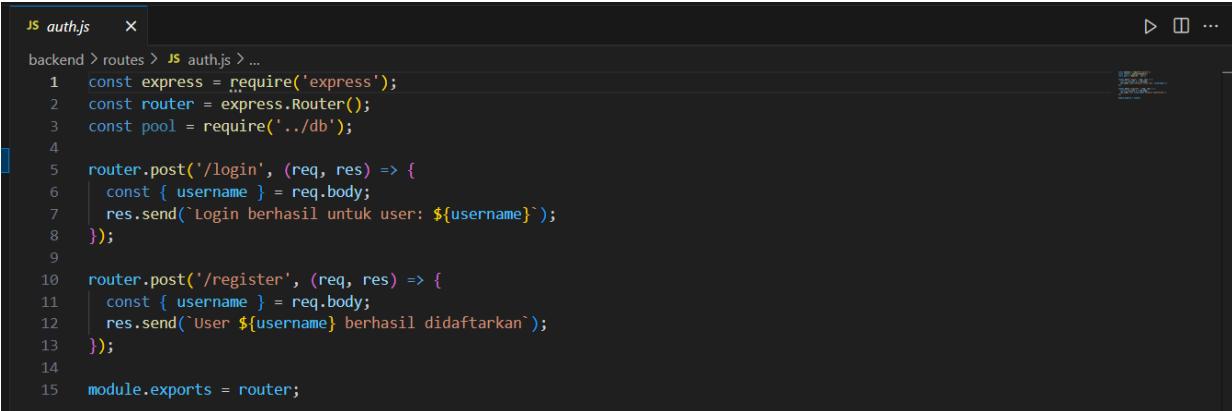
Sekumpulan endpoint yang digunakan untuk mengelola data tempat kost dalam aplikasi—baik untuk pemilik kost.

- **Endpoint Keuangan:**

```
JS keuangan.js x
backend > routes > JS keuangan.js > ...
1 const express = require('express');
2 const router = express.Router();
3 const pool = require('../db');
4
5 router.get('/', (req, res) => {
6   res.send('Data keuangan berhasil diambil');
7 });
8
9 router.post('/', (req, res) => {
10   const { type, amount } = req.body;
11   res.send(`Data keuangan (${type}, ${amount}) berhasil ditambahkan`);
12 });
13
14 module.exports = router;
```

API yang digunakan untuk mengelola data transaksi dan aktivitas keuangan dalam sistem coffee shop

- **Endpoint Autentikasi:**



```
JS auth.js ×
backend > routes > JS auth.js > ...
1 const express = require('express');
2 const router = express.Router();
3 const pool = require('../db');
4
5 router.post('/login', (req, res) => {
6   const { username } = req.body;
7   res.send(`Login berhasil untuk user: ${username}`);
8 });
9
10 router.post('/register', (req, res) => {
11   const { username } = req.body;
12   res.send(`User ${username} berhasil didaftarkan`);
13 });
14
15 module.exports = router;
```

API yang mengatur proses login, register, dan pengamanan akses pengguna. Ini adalah "gerbang masuk" ke sistem, biar hanya user yang terdaftar dan valid yang bisa akses fitur tertentu.

---