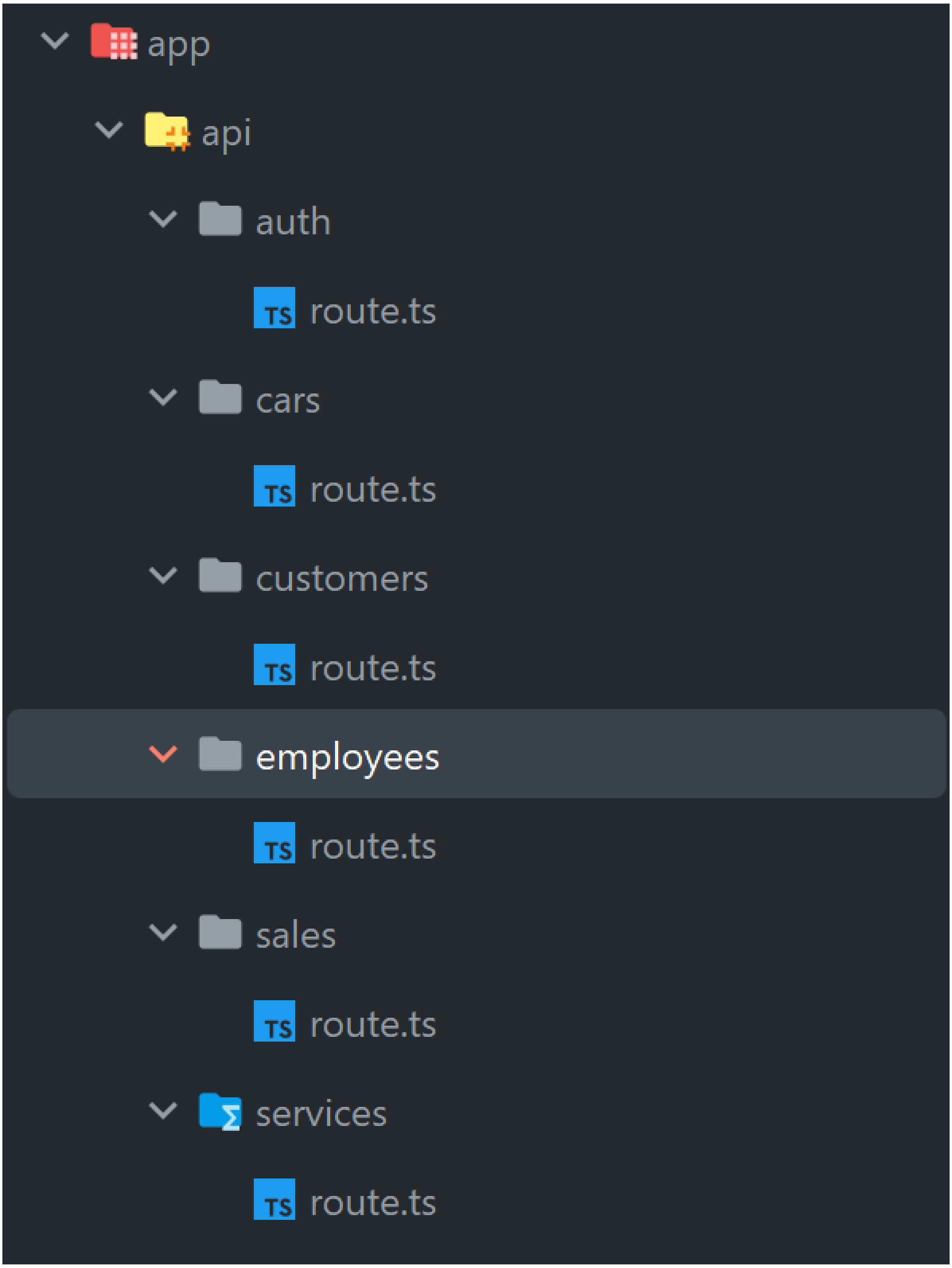
Лабораторна робота №3

Для реалізації REST API в обробники було додано усі відповідні CRUD операції



```
app
  api
    auth
      route.ts
    cars
      route.ts
    customers
      route.ts
    employees
      route.ts
    sales
      route.ts
    services
      route.ts
```

```
import {sql} from "@vercel/postgres";
import {NextResponse} from "next/server";
import jwt from "jsonwebtoken";
import {cookies} from "next/headers";

export async function POST(formData) {
   try {
       const login = formData.get('login');
       const password = formData.get('password');

       const result = await sql`SELECT id, password
                                FROM Employees
                                WHERE lastname = ${login};`;
       if (result.rows.length === 0) {
           throw new Error("Undefined user");
       }

       const user = result.rows[0];
       if (!(password === user.password)) {
           throw new Error("Incorrect password");
       }

       const token = jwt.sign({userId: user.id}, process.env.SECRET_KEY,
{expiresIn: '1h'});

       cookies().set('Authorization', 'Bearer ' + token)

       return NextResponse.json({token});
   } catch (e) {
       return NextResponse.json({error: e})
   }
}



import {sql} from "@vercel/postgres";
import {revalidatePath} from "next/cache";

export async function GET() {
   return Response.json((await sql`SELECT *
                                   FROM Cars;`).rows)
}

export async function POST(formData: FormData) {
   try {
       await sql`INSERT INTO Cars (Brand, Model, Year, Price)
                 VALUES (${formData.get('brand')},
                         ${formData.get('model')},
                         ${formData.get('year')},
                         ${formData.get('price')});
       `
       revalidatePath('/cars');
       return Response.json({success: true});
   } catch (e) {
       return Response.json({success: true});
   }
}

export async function PATCH(formData: FormData) {
   try {
       await sql`UPDATE Cars
                 SET Brand = ${formData.get('brand')},
                     Model = ${formData.get('model')},
                     Year  = ${formData.get('year')},
                     Price = ${formData.get('price')}
                 WHERE id = ${formData.get('id')};
       `
       return Response.json({success: true});
   } catch (e) {
       return Response.json({success: false});
   }
}

export async function DELETE(formData: FormData) {
   try {
       await sql`DELETE
                 FROM Cars
                 WHERE id = ${formData.get('id')};`
       return Response.json({success: true});
   } catch (e) {
       return Response.json({success: false});
   }
}
```

```
import {sql} from "@vercel/postgres";
import {revalidatePath} from "next/cache";

export async function GET() {
   return Response.json((await sql`SELECT *
                                   FROM Customers;`).rows)
}

export async function POST(formData: FormData) {
   try {
       await sql`INSERT INTO Customers (FirstName, LastName, Email, Phone)
                 VALUES (${formData.get("firstname")},
                         ${formData.get("lastname")},
                         ${formData.get("email")},
                         ${formData.get("phone")});
       `;
       return Response.json({success: true});
   } catch (e) {
       return Response.json({success: false});
   }
}

export async function PATCH(formData: FormData) {
   try {
       await sql`UPDATE Customers
                 SET FirstName = ${formData.get("firstname")},
                     LastName  = ${formData.get("lastname")},
                     Email     = ${formData.get("email")},
                     Phone     = ${formData.get("phone")}
                 WHERE id = ${formData.get("id")};
       `;
       return Response.json({success: true});
   } catch (e) {
       return Response.json({success: false});
   }
}

export async function DELETE(formData: FormData) {
   try {
       await sql`DELETE
                 FROM Customers
                 WHERE id = ${formData.get("id")};`;
       return Response.json({success: true});
   } catch (e) {
       return Response.json({success: false});
   }
}
```

```
import {sql} from "@vercel/postgres";
import {revalidatePath} from "next/cache";

export async function GET() {
   return Response.json((await sql`SELECT *
                                   FROM Employees;`).rows)
}

export async function POST(formData: FormData) {
   try {
       await sql`INSERT INTO Employees (Firstname, Lastname, Position, Salary,
Password)
                 VALUES  (${formData.get('firstname')},
                         ${formData.get('lastname')},
                         ${formData.get('position')},
                         ${formData.get('salary')},
                         ${formData.get('password')});
       `;
       return Response.json({success: true});
   } catch (e) {
       return Response.json({success: false});
   }
}

export async function PATCH(formData: FormData) {
   try {
       await sql`UPDATE Employees
                 SET Firstname = ${formData.get('firstname')},
                     Lastname  = ${formData.get('lastname')},
                     Position  = ${formData.get('position')},
                     Salary    = ${formData.get('salary')},
                     Password  = ${formData.get('password')}
                 WHERE id = ${formData.get('id')};
       `;
       return Response.json({success: true});
   } catch (e) {
       return Response.json({success: false});
   }
}

export async function DELETE(formData: FormData) {
   try {
       await sql`DELETE
                 FROM Employees
                 WHERE id = ${formData.get('id')};`;
       return Response.json({success: true});
   } catch (e){
       return Response.json({success: false});
   }
}
```

```
import {sql} from "@vercel/postgres";
import {revalidatePath} from "next/cache";

export async function GET() {
   return Response.json((await sql`SELECT *
                                   FROM Sales;`).rows);
}

export async function POST(formData: FormData) {
   try {
       await sql`INSERT INTO Sales (CarID, CustomerID, SaleDate, SalePrice)
                 VALUES (${formData.get('carid')},
                         ${formData.get('customerid')},
                         ${formData.get('saledate')},
                         ${formData.get('saleprice')});
       `;
       return Response.json({success: true});
   } catch (e) {
       console.log(e)
       return Response.json({success: false});
   }
}

export async function PATCH(formData: FormData) {
   try {
       await sql`UPDATE Sales
                 SET CarID      = ${formData.get('carid')},
                     CustomerID = ${formData.get('customerid')},
                     SaleDate   = ${formData.get('saledate')},
                     SalePrice  = ${formData.get('saleprice')}
                 WHERE Id = ${formData.get('id')};
       `;
       return Response.json({success: true});
   } catch (e) {
       return Response.json({success: false});
   }
}

export async function DELETE(formData: FormData) {
   try {
       await sql`DELETE
```

```
                 FROM Sales
                 WHERE Id = ${formData.get('id')};
       `;
       return Response.json({success: true});
   } catch (e) {
       return Response.json({success: false});
   }
}




import {sql} from "@vercel/postgres";
import {revalidatePath} from "next/cache";

export async function GET() {
   return Response.json((await sql`SELECT *
                                   FROM Services;`).rows);
}

export async function POST(formData) {
   try {
       await sql`INSERT INTO Services (Service_name, Description, Price)
                 VALUES (${formData.get('service_name')},
                         ${formData.get('description')},
                         ${formData.get('price')};
       `;
       return Response.json({success: true});
   } catch (e) {
       console.log(e)
       return Response.json({success: false});
   }
}

export async function PATCH(formData) {
   try {
       await sql`UPDATE Services
                 SET Service_Name = ${formData.get('service_name')},
                     Description  = ${formData.get('description')},
                     Price        = ${formData.get('price')}
                 WHERE id = ${formData.get('id')};
       `;
       return Response.json({success: true});
   } catch (e) {
       return Response.json({success: false});
   }
}

export async function DELETE(formData) {
   try {
       await sql`DELETE
                 FROM Services
                 WHERE id = ${formData.get('id')};
       `;
       return Response.json({success: true});
   } catch (e) {
       return Response.json({success: false});
   }
```