project-root/

├── common/

│ └── databaseConnection.ts

├── modules/

│ └── user/

│ ├── user.controller.ts

│ └── user.routes.ts

├── src/

│ └── server.ts

├── tests/

│ └── user.test.ts

├── \_\_mocks\_\_/

│ └── common/

│ └── databaseConnection.ts

├── jest.config.js

├── package.json

// common/databaseConnection.ts

export const db = {

query: async (sql: string, params: any[]) => {

// Simulate real DB logic (not used in tests)

}

};

// modules/user/user.controller.ts

import { Request, Response } from 'express';

import { db } from '../../common/databaseConnection';

export const getUsers = async (req: Request, res: Response) => {

try {

const result = await db.query('SELECT \* FROM users', []);

res.status(200).json(result.rows);

} catch (error) {

res.status(500).json({ message: 'Error fetching users' });

}

};

// modules/user/user.routes.ts

import express from 'express';

import { getUsers } from './user.controller';

const router = express.Router();

router.get('/', getUsers);

export default router;

// src/server.ts

import express from 'express';

import userRoutes from '../modules/user/user.routes';

const app = express();

app.use(express.json());

app.use('/user', userRoutes);

const PORT = process.env.PORT || 3000;

app.listen(PORT, () => {

console.log(`Server running on port ${PORT}`);

});

export default app;

// \_\_mocks\_\_/common/databaseConnection.ts

export const db = {

query: jest.fn(),

};

// tests/user.test.ts

import request from 'supertest';

import app from '../src/server';

import { db } from '../common/databaseConnection';

jest.mock('../common/databaseConnection');

describe('User Routes', () => {

beforeEach(() => {

jest.clearAllMocks();

});

it('GET /user returns list of users', async () => {

(db.query as jest.Mock).mockResolvedValueOnce({

rows: [{ id: 1, name: 'John Doe' }],

});

const res = await request(app).get('/user');

expect(res.status).toBe(200);

expect(res.body).toEqual([{ id: 1, name: 'John Doe' }]);

});

});

// jest.config.js

module.exports = {

preset: 'ts-jest',

testEnvironment: 'node',

moduleFileExtensions: ['ts', 'js'],

roots: ['<rootDir>/tests'],

};

// package.json (relevant parts only)

{

"scripts": {

"test": "jest"

},

"devDependencies": {

"jest": "^29.0.0",

"ts-jest": "^29.0.0",

"@types/jest": "^29.0.0",

"@types/supertest": "^2.0.12",

"supertest": "^6.3.0",

"typescript": "^4.5.5"

},

"dependencies": {

"express": "^4.18.2"

}

}