**Step 1: Setting up the Nodejs with Yarn in the Windows**

Installed Nodejs from the official website and checked the version of Nodejs in Command Prompt

A screen shot of a computer

Description automatically generated

A black background with white text

Description automatically generated

Npm config set fund false --location=global

A screenshot of a computer screen

Description automatically generated

§ cd banking-backend

– To locate to the folder

Visual Studio Code 🡪 Open Folder 🡪 Redirect to the Project installation folder 🡪 open Terminal in VS Code and run ‘yarn install’ to installs all dependencies listed in the package.json file.

$yarn run start

Navigate to url <http://localhost:3000>

**Step 2 Installed Necessary Package in the VS Code Extension**

Prettier for code formatting.

ESLint for linting.

Docker for managing Docker containers.

NestJS Essential Extension Pack

**Step 3: Install required dependencies for PostgreSQL, TypeORM, and GremlinJS**



npm install @nestjs/typeorm typeorm pg gremlin

 @nestjs/typeorm: NestJS integration for TypeORM.

 typeorm: ORM for TypeScript and JavaScript.

 pg: PostgreSQL driver for Node.js.

 gremlin: GremlinJS client for interacting with graph databases.

**Step 4: add docker-compose.yml file to the root of the project with the following code and created the database with the following data.**

*A screenshot of a computer

Description automatically generated*

Run the below code in terminal to run the docker container

$docker-compose up

**Step 5: import connection string and necessary entities to app.module.ts**

A computer screen shot of a program

Description automatically generated

**Step 6 Created Entities for Person, Transaction, BankAccount, Friendship, MaxBorrowing in the respective folder and added DTO files for creating and updating into the database.**

**Step 7 Executed the code below code for adding functionalities and performed the CRUD operation to the entities**

$nest generate controller person

$nest generate service person

$nest generate module person

$nest generate controller bank-account

$nest generate service bank-account

$nest generate module bank-account

$nest generate controller transaction

$nest generate service transaction

$nest generate module transaction

$nest generate controller friendship

$nest generate service friendship

$nest generate module friendship

**Step 8: Run and Test the code using Swagger**

Executed:

$yarn add @nestjs/swagger swagger-ui-express

Added the below code in main.ts for configuring in the Project

**A screenshot of a computer code

Description automatically generated**

Test the code by redirecting to the default url.

<http://localhost:3000/api>

**Step 9: Executed the code below code for adding main functionalities to the Project**

$nest generate controller process

$nest generate service process

$nest generate module process

Added functionalities such as update balance, Calculating Net Worth and Max borrow limit

**Main Requirements of the Project**

1. Find the new balance for each bank account using the new transactions in the database that were not present the last time it was retrieved.

2. Calculate each person's net worth -sum of the balances of all their bank accounts.

3. Calculate the maximum amount a person can borrow from his friends.

**Step 10: Below steps to schedule the process to run at nighttime exactly at 12:00**

Executed

$yarn add @nestjs/schedule

$yarn add @nestjs/cron

$nest generate service nightly-process

$nest generate module nightly-process

A screenshot of a computer program

Description automatically generated

@Cron('0 0 \* \* \*') decorator sets the job to run at midnight daily

**Step 11: Added unit test case in all spec file and verified all the unit test cases**

**Step 12: Mock Data for executing the file**

Person

**{**

**"name": "Dinesh Ram",**

**"email": "dinesh@example.com"**

**}**

Bank Account

**{**

**"iban": "DE89370400440532013006",**

**"balance": 10000,**

**"person\_id": 1**

**}**

Transaction

**{**

**"sender\_iban": "DE89370400440532013000",**

**"receiver\_iban": "DE89370400440532013002",**

**"amount": 500**

**}**

Friendship

**{**

**"person\_id": 1,**

**"friend\_id": 2**

**}**

Webhook Process

**{**

**"processId":3,**

**"personId":1**

**}**