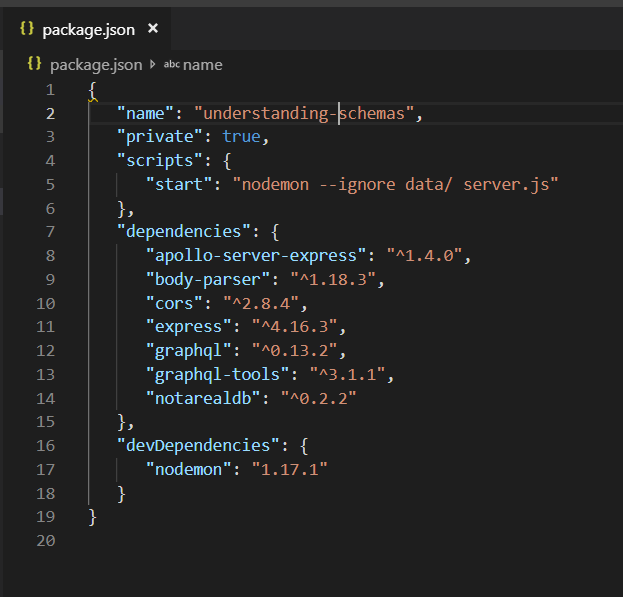
Let us create a simple application to understand this schema.

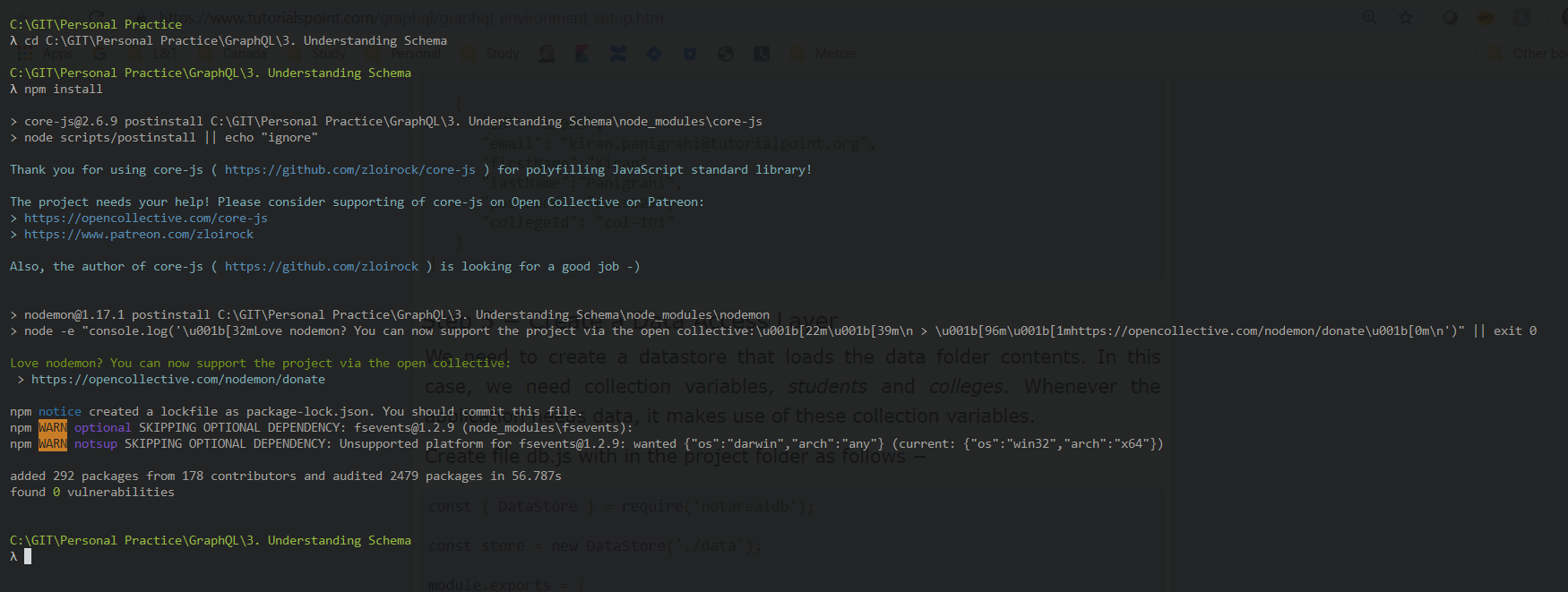
**This will create a schema for querying list of students from the server**.

The student data will be stored in a flat file and **we will use a node module called notarealdb to fake a database and read from the flat file**.

## Step 1 − Download and Install Required Dependencies for the Project

Create new package.json file with below code and install all dependencies.

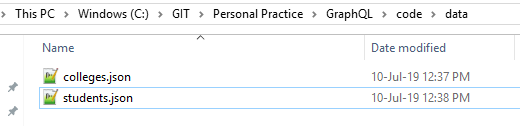


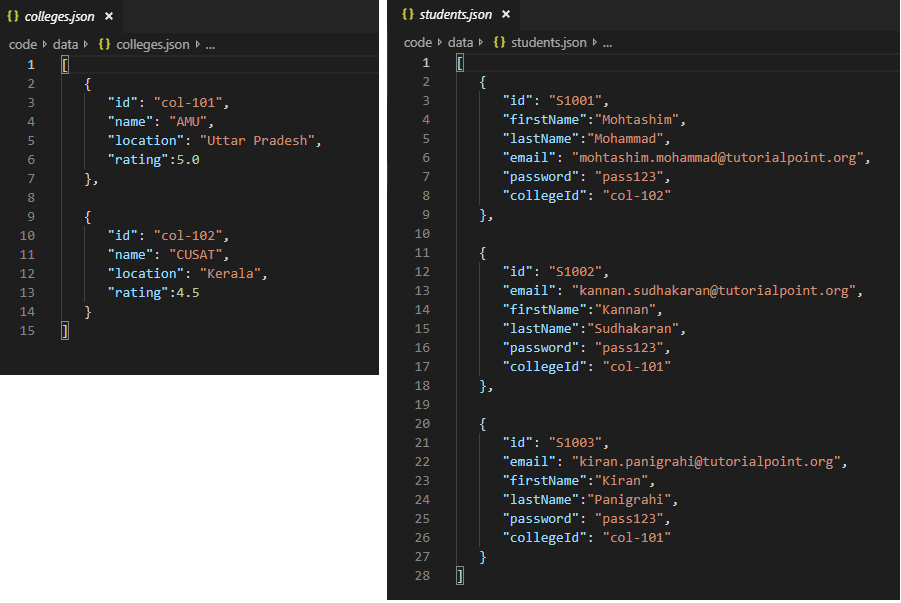


**Create Flat File Database in Data Folder**

we use flat files to store and retrieve data.

Create a folder data and add two files **students.json** and **colleges.json**.





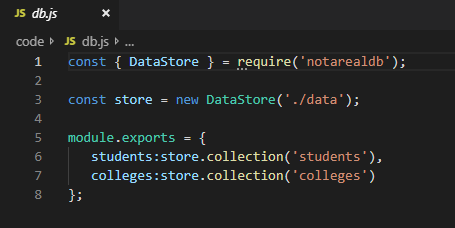
**Create a Data Access Layer**

We need to create a datastore that loads the data folder contents.

In this case, we need collection variables, *students* and *colleges*.

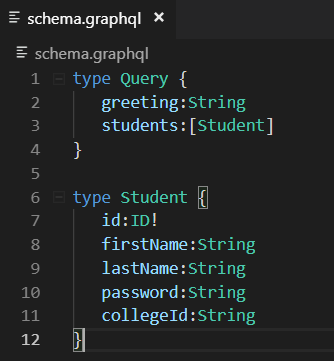
Whenever the application needs data, it makes use of these collection variables.

Create file db.js with in the project folder as follows



## Step 2 − Create a Schema

Add **schema.graphql** file and code to the project folder



The root of the schema will be Query type.

The query has two fields –

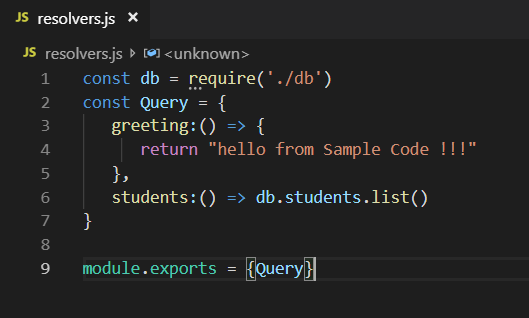
* greeting (returns String) and
* students (returns a list of students).

Student is declared as an Object type since it contains multiple fields.

The ID field is declared as non-nullable.

## Step 3 − Create Resolver

Add **resolvers.js** file and code to the project folder



Here greeting and students are the resolvers that handle the query.

**students resolver function** returns a list of students from the data access layer.

To access resolver functions outside the module, Query object has to be exported using **module.exports**.

## Step 4 − Run the Application

Create a server file and configure GraphQL as follows

