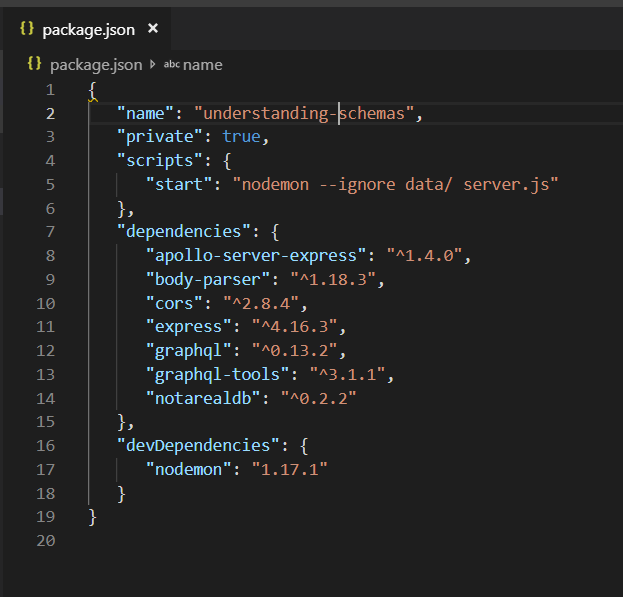
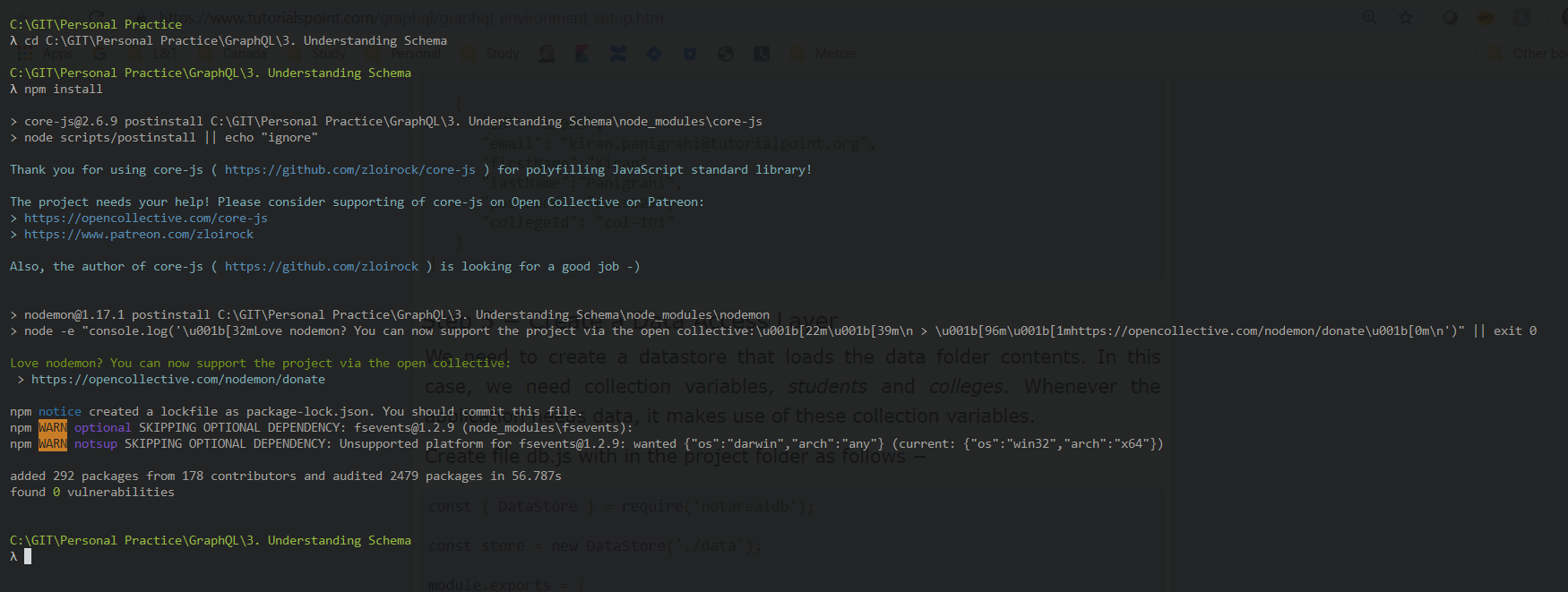
This will create schema for querying a student by id from the server.

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 flat file.

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

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

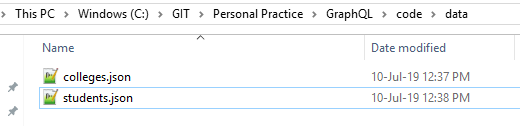


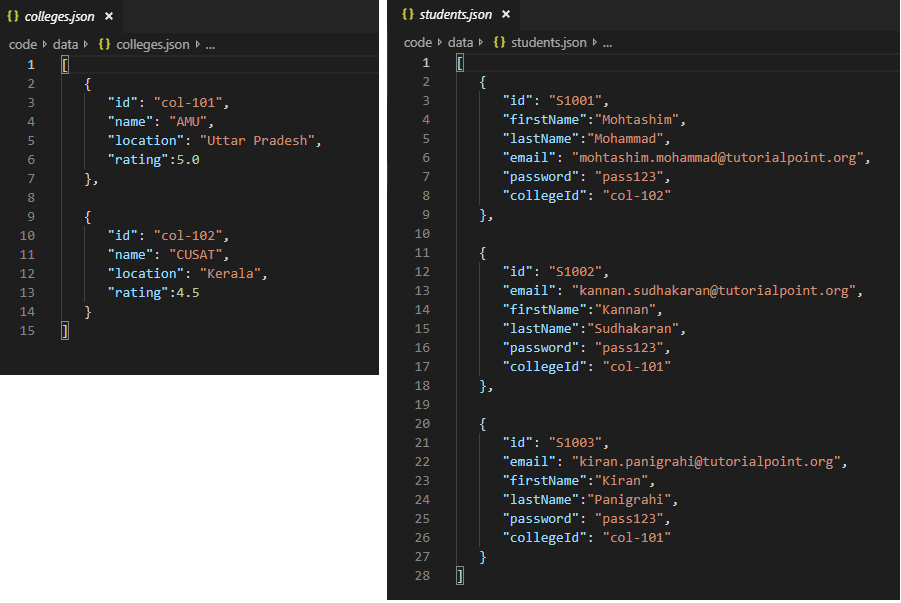


**Create Flat File Database in Data Folder**

we use flat files to store and retrieve data.

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





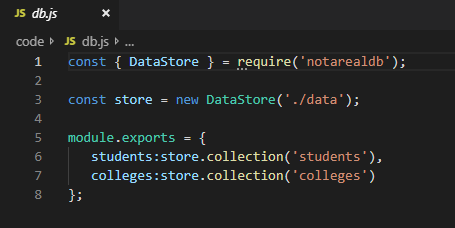
**Create a Data Access Layer**

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

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

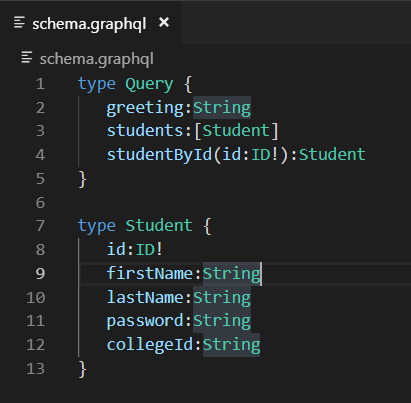
Whenever the application needs data, it makes use of these collection variables.

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



## Step 2 − Create a Schema

Add schema.graphql file with following code



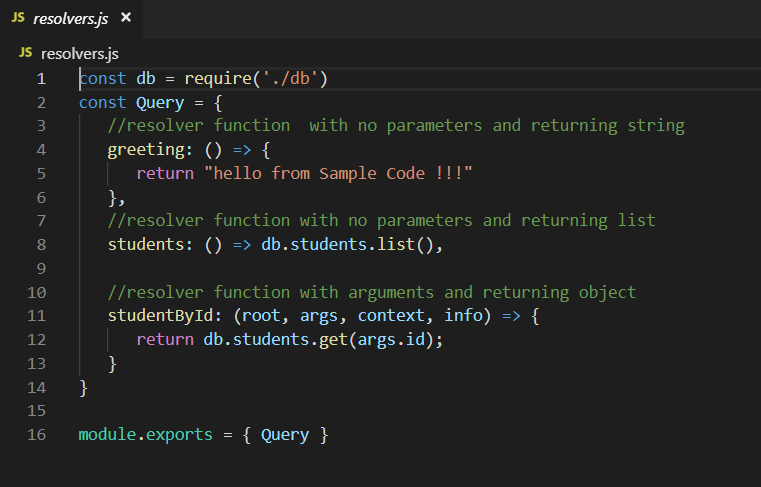
The schema file shows that user can query for

* *greeting,*
* *students* and
* *studentById*.

To retrieve students with specific id, we use **data type ID!** which shows a non nullable unique identifier field.

The *students* field returns an array of students, and *greeting* returns a simple string value.

## Step 3 − Create Resolver



Here, *studentById* takes in three parameters.

* the *studentId* can be retrieved from args;
* root will contain the Query object itself.

To return a specific student, we need to call get method with id parameter in the students collection.

Here ***greeting, students, studentById* 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.js file.

