**LAB # 5**

**Intermediate Java script**

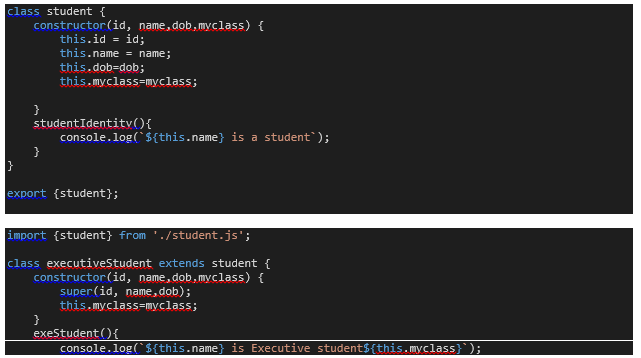
**OBJECTIVE**

To get familiar and become more knowledgeable in implementing real world JS use cases

**Lab Task**

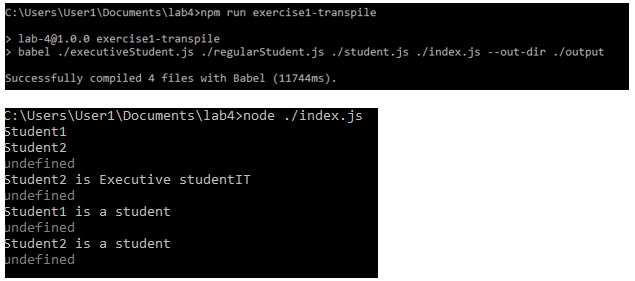
1. Classes and Inheritance:
   1. Create a base class called *Student* and export it
      1. Add id, name, date of birth properties
      2. Add *enroll* method
   2. Create two child classes called *RegularStudent* and *ExecutiveStudent* and inherit them from *Student* base class
      1. Add *attendLab* method in *RegularStudent* class
      2. Add *attendTheory* method in *ExecutiveStudent* class
   3. Create a separate module and import *RegularStudent* and *ExecutiveStudent* classes. Validate using *console.log* that the properties from base classes have been inherited into child classes along with own properties.

**Input:**



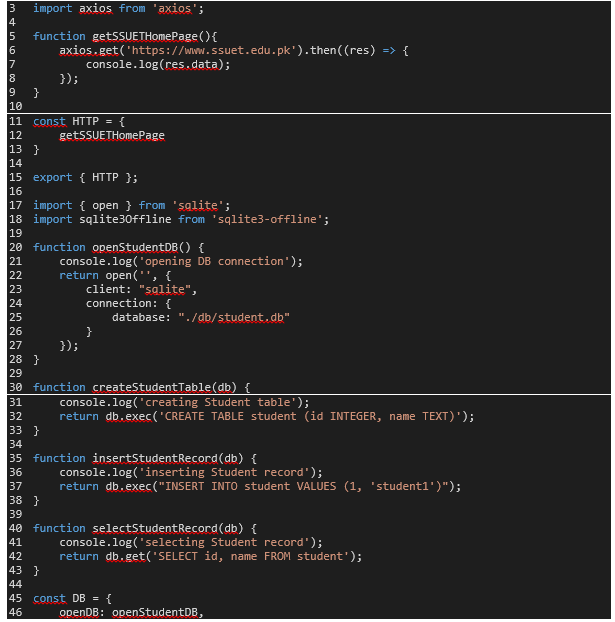


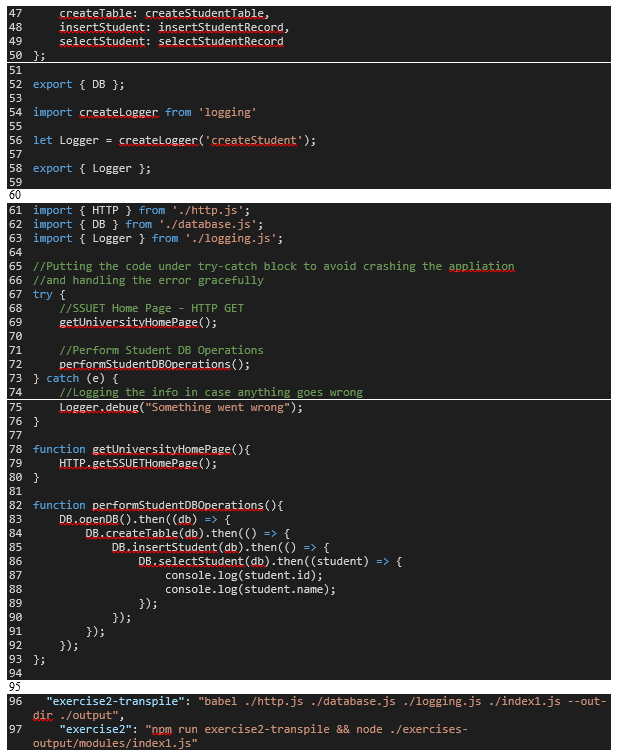
**Output:**



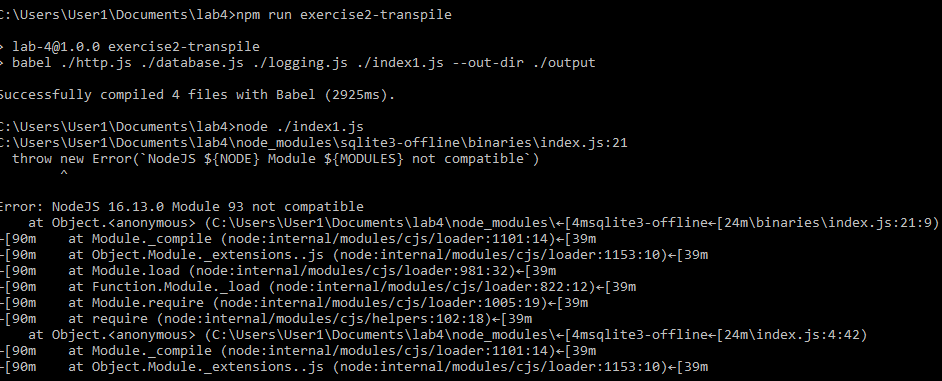
1. HTTP and Database:
   1. Import *sqlite* and *axios* packages in separate module files
   2. Create DB, open connection and perform CRUD operations for *Student using sqlite* library
      1. *Create* and open database
      2. *Create* Student table using properties from *Student* class created above
      3. *Insert* two records for both regular and executive student
      4. *Select* all records
   3. Fetch the university *About* page using HTTP *axios* library
      1. Using *GET* method of axios library, fetch SSUET website *About* page

**Input:**



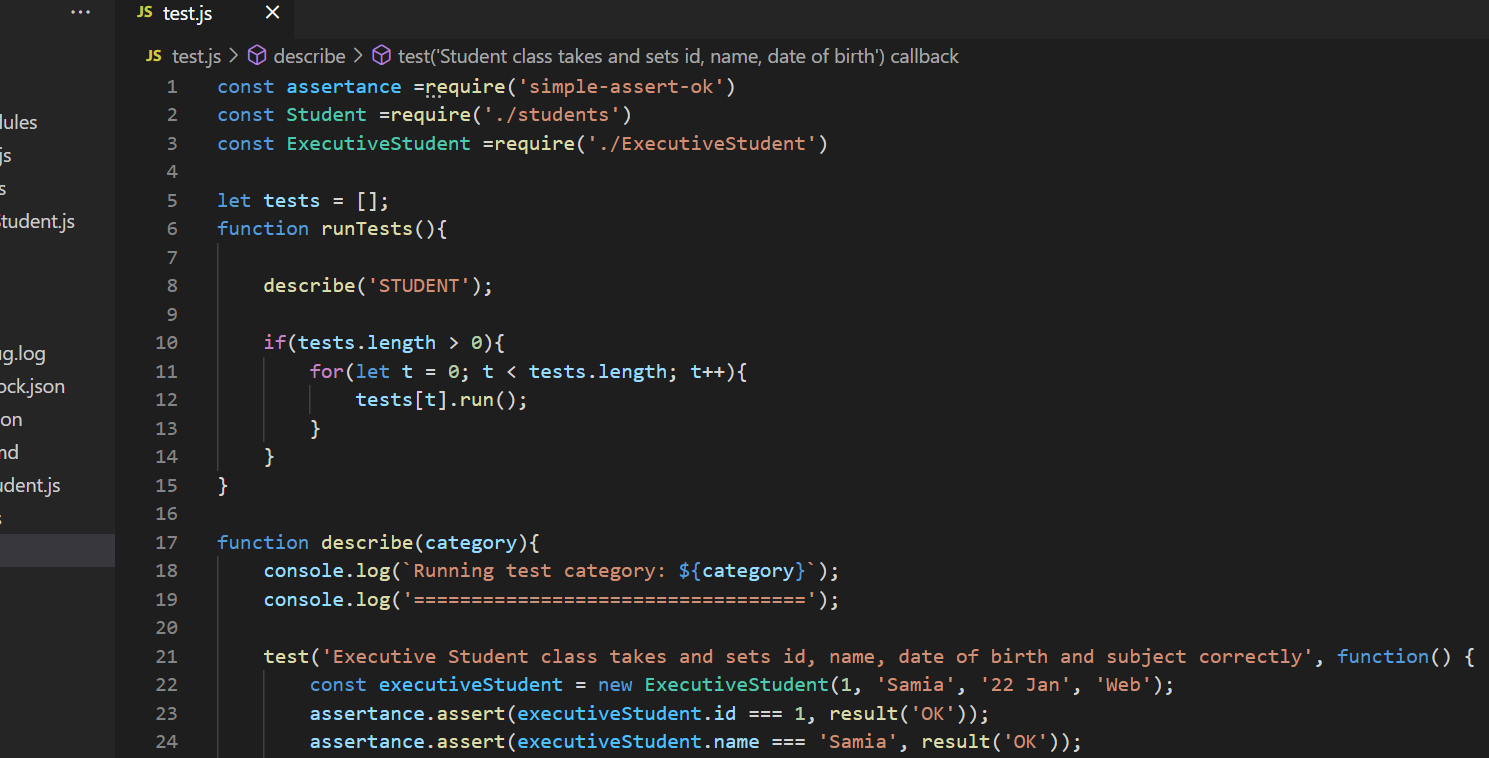


**Output:**



1. Testing:
   1. Test *Student* class
      1. Use the example above to copy paste and validate Student properties
   2. Test either *RegularStudent* or *ExecutiveStudent* class
      1. Use the example above to copy paste and validate either *RegularStudent* or *ExecutiveStudent* class.

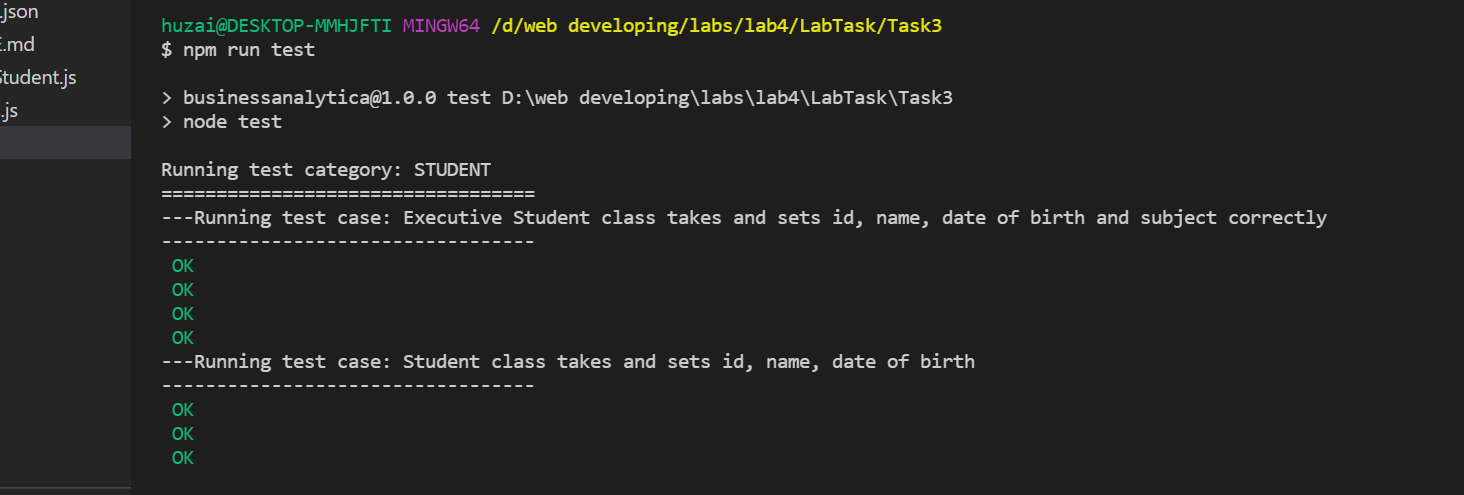
**Input:**







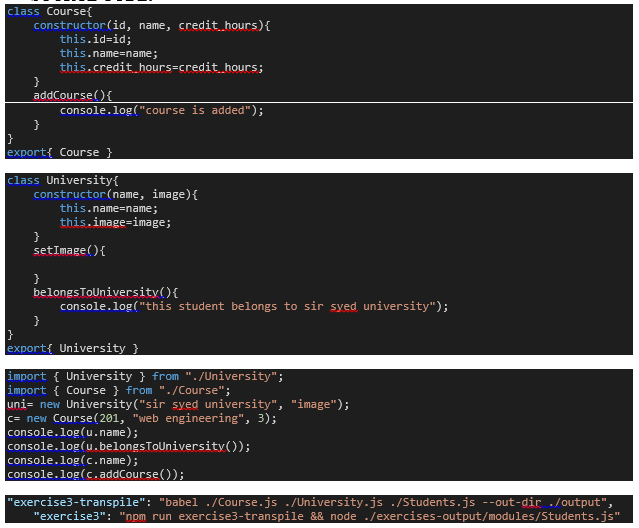
**Output:**



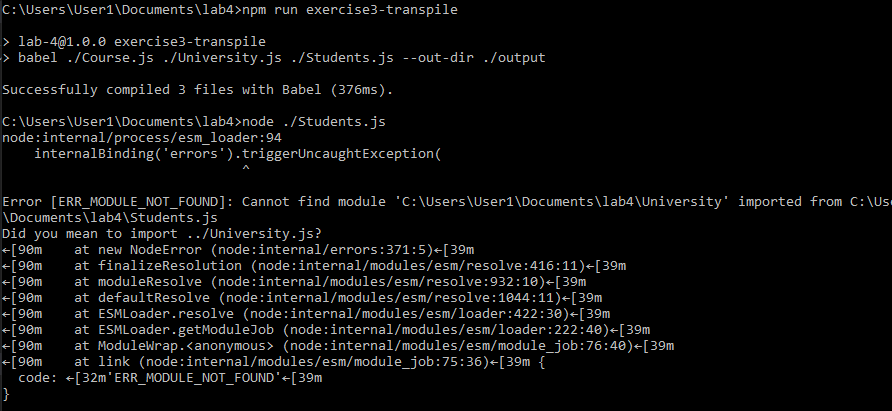
**Home Task**

1. Classes and Inheritance:
   1. Create a class called *Course* and export it
      1. Add id, name, credit hours properties
   2. Create a class called *University* and export it
      1. Add name, image properties
      2. Add a method called *setImage*
   3. Modify the *Student* class
      1. Add a property called *university* of type *University* class
      2. Add a property called *courses* of an array of *Course* class
      3. Add a method called *addCourse* accepting *Course* class object and set that onto *courses* property
      4. Add a method called *belongsToUniversity* accepting *University* class object and set that onto *university* property

**Input:**

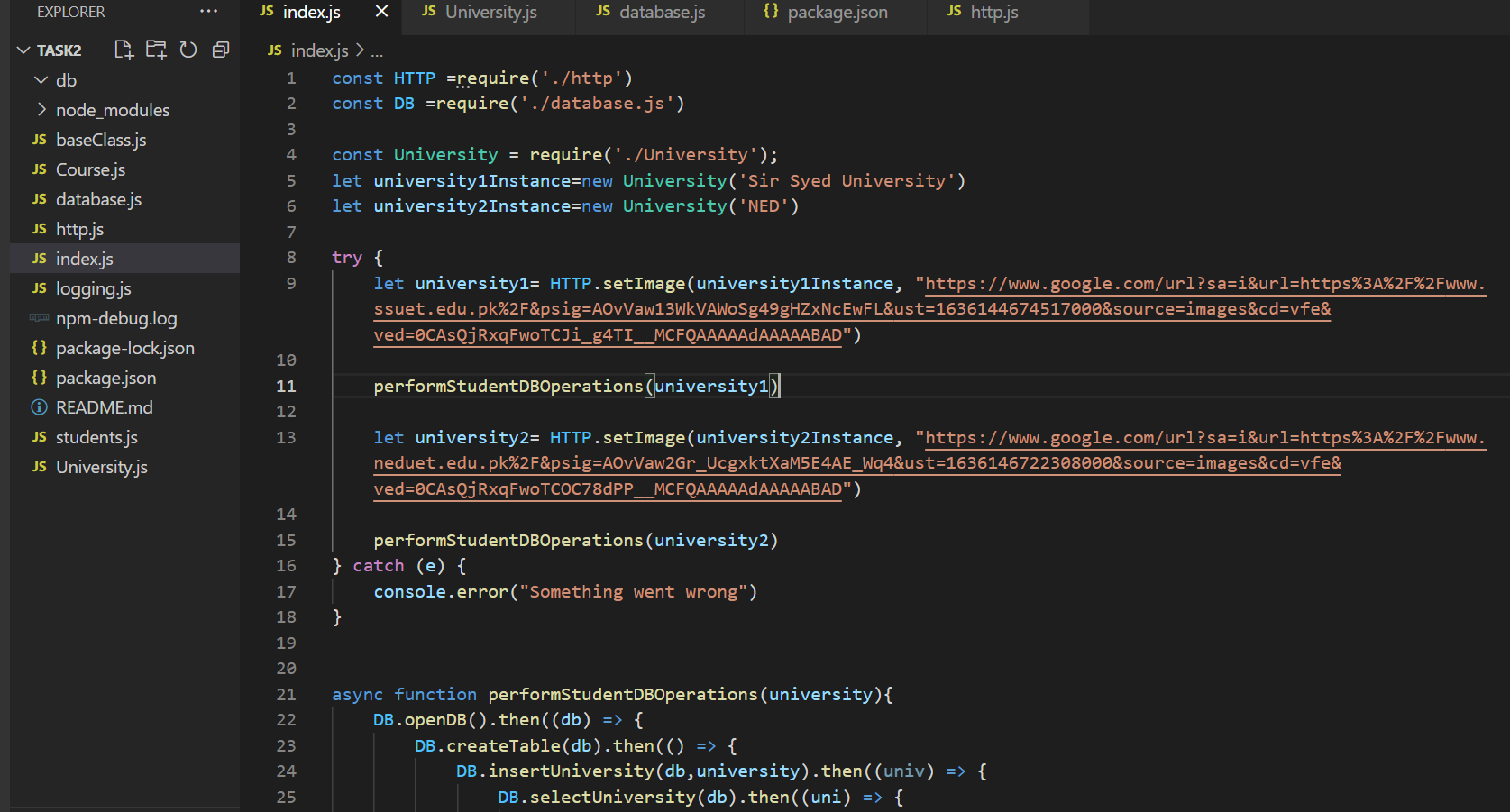
****

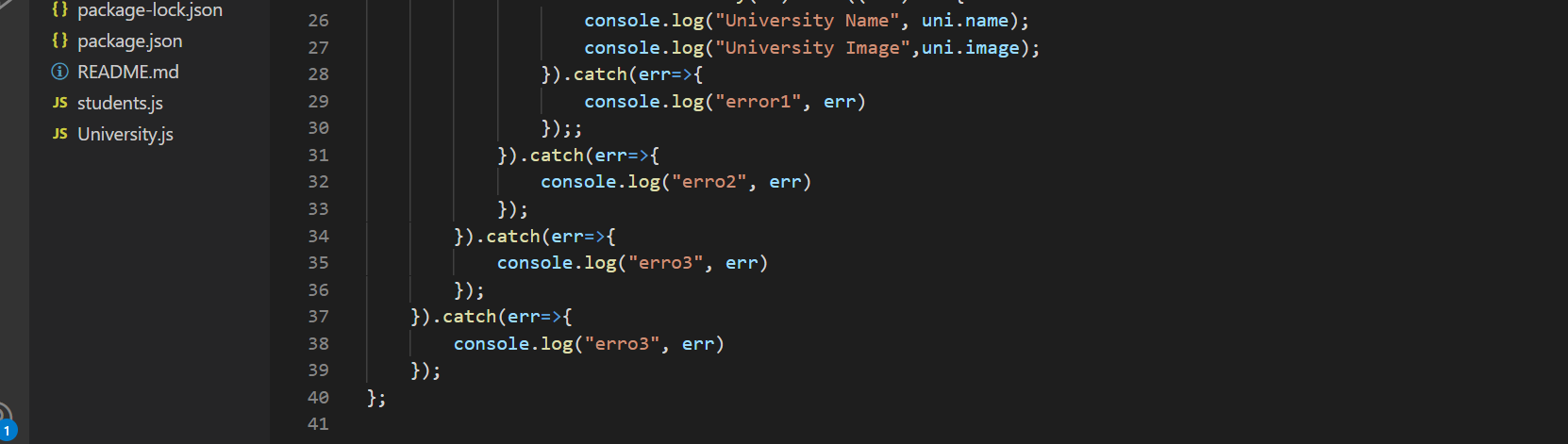
**Output:**

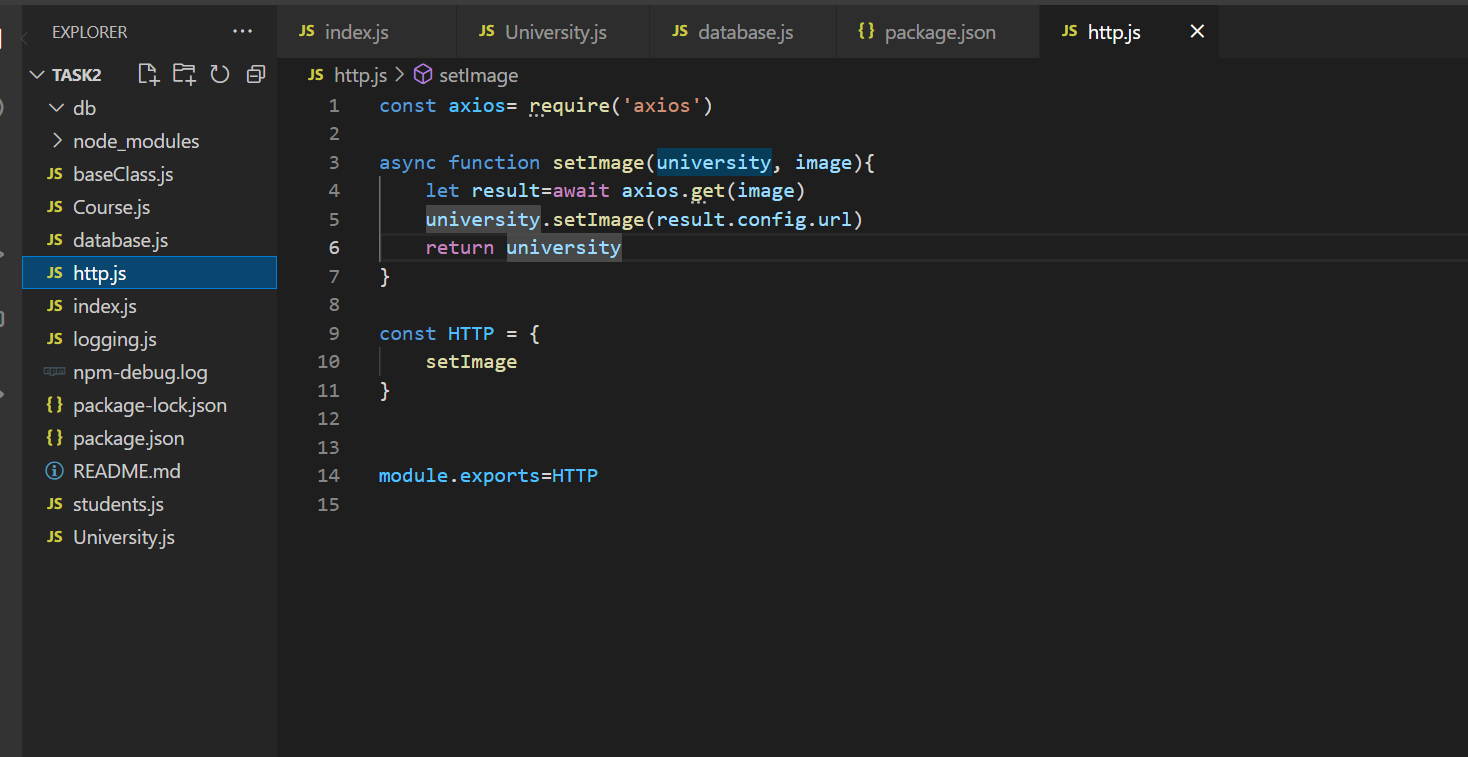


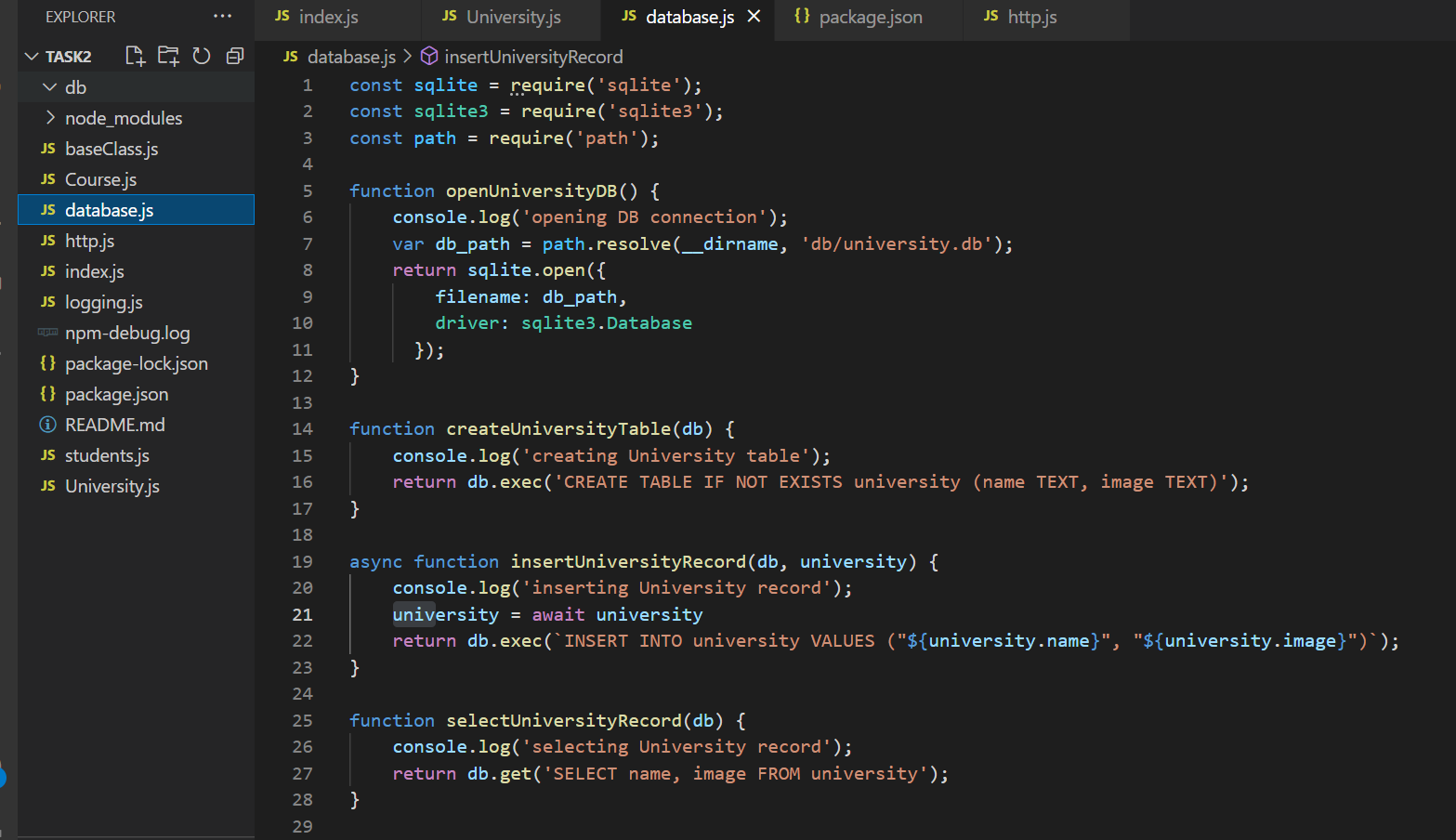
1. HTTP and Database:
   1. Modify the University class’s *setImage* method
      1. Fetch the SSUET logo from the SSUET website using *axios* *GET* method and set it to *image* property
      2. Create a new instance and call *setImage* method on the instance and validate using console.log if the image property has been set
   2. Create DB, open connection and perform CRUD operations for *University* using *sqlite* library
      1. *Create* and open database
      2. *Create* University table using properties from *University* class created above
      3. *Create* two *University* instances
      4. *Call* *setImage* method on both the instances
      5. *Insert* both the instances using their *name* and *image* properties

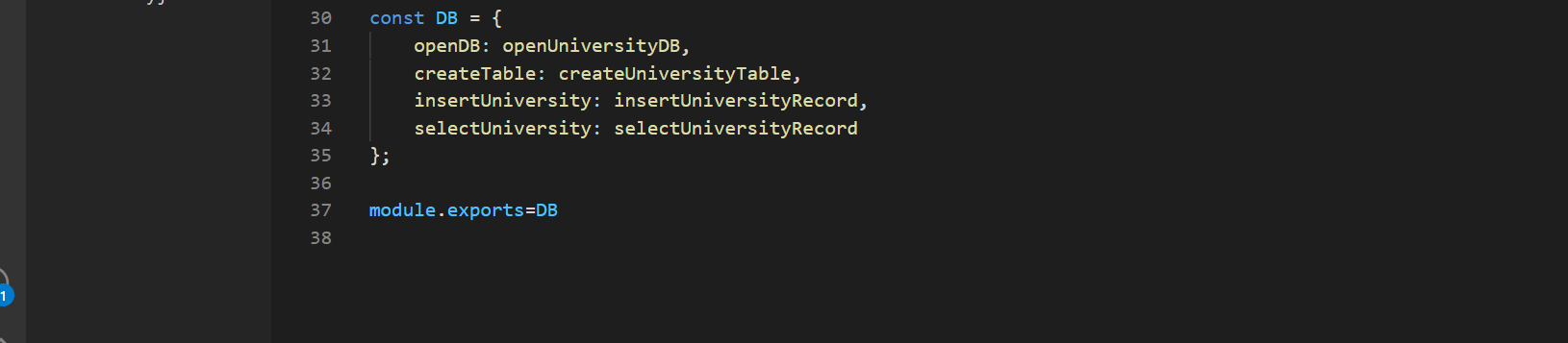
**Input:**



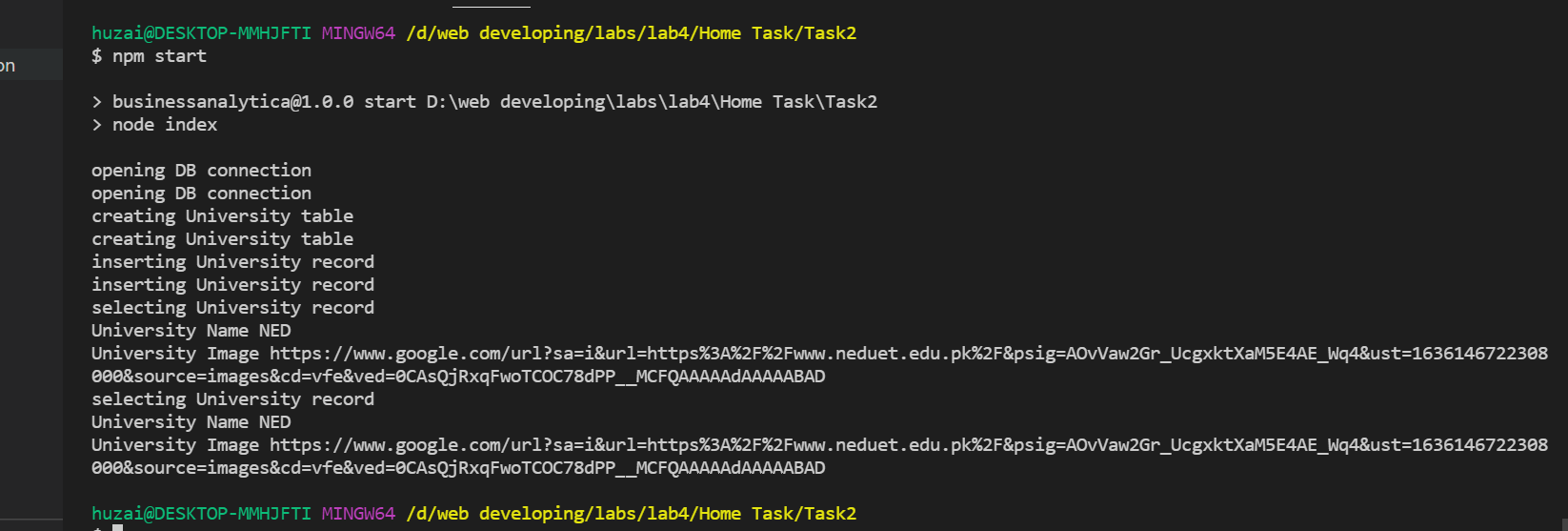


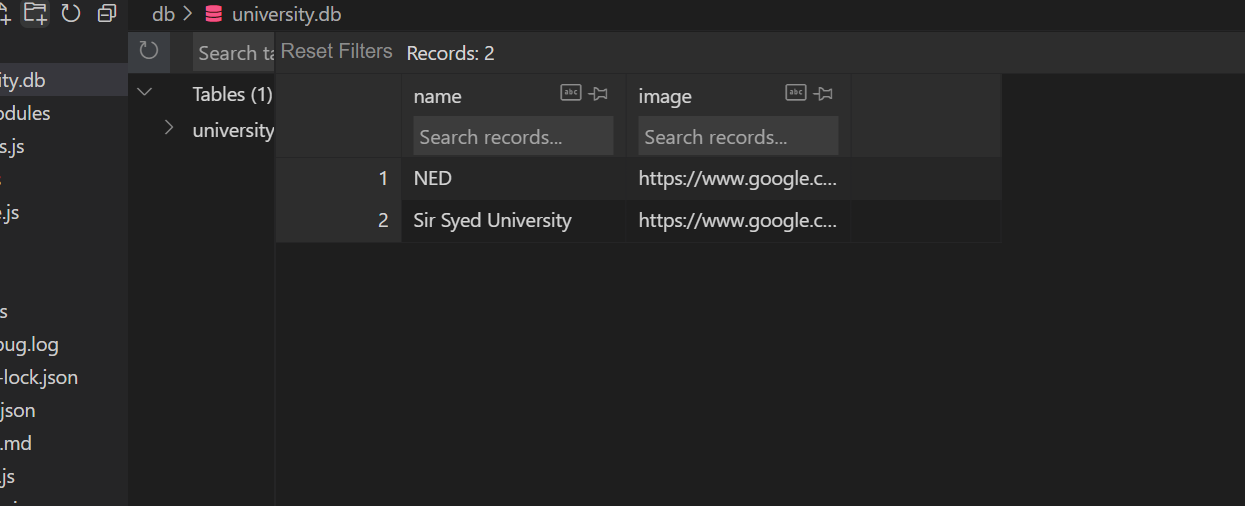






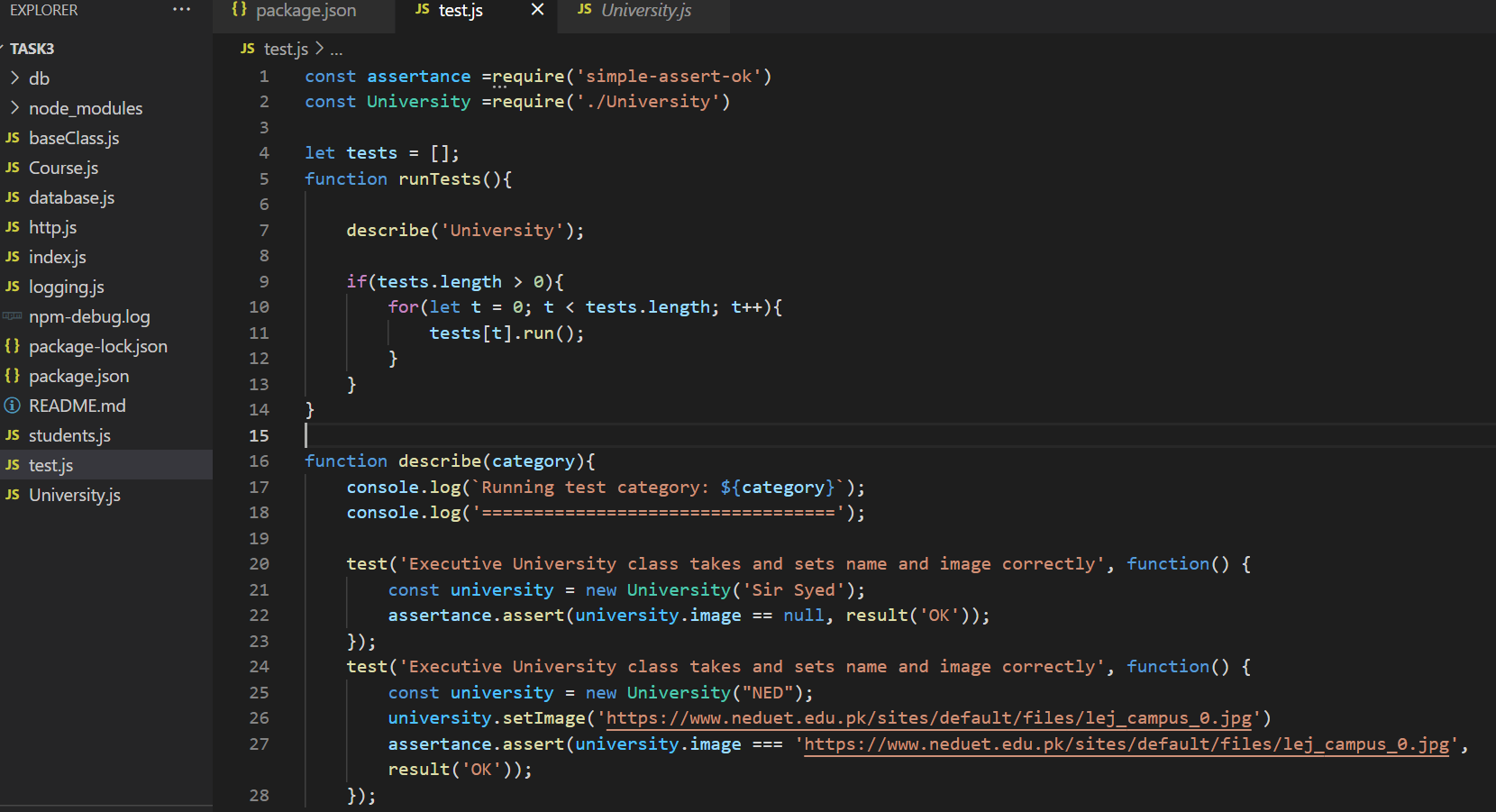
**Output:**





1. Testing:
   1. Test *University* class.
      1. Create a *University* instance
      2. Assert if the instance has image property set to empty string
      3. Call *setImage* method on the new instance
      4. Assert if the instance has image property set to some string

**Input:**





**Output:**

