**1. app.py:**

* This file is the entry point of the Flask application.
* It creates and configures the Flask app.
* The Flask class is imported from the flask module, and an instance of the Flask class is created as app.
* The SQLAlchemy database URI is set in the app's configuration to specify the connection details for the PostgreSQL database.
* The db instance from the database module is initialized with the app using db.init\_app(app).
* A simple route is defined for the root URL ("/") that returns the string "Student Management System" when accessed.
* The shutdown\_session function is registered as a teardown function to remove the database session at the end of each request.
* Finally, the GraphQL endpoint is added to the app using app.add\_url\_rule with the GraphQLView from Strawberry.

**2. dao.py:**

* This file contains the Data Access Object (DAO) for the Student model.
* The DAO provides methods to interact with the database and perform CRUD operations related to students.
* It imports the Student model from the models module and the db instance from the database module.
* The methods in the StudentDAO class include:
* get\_all\_students(): Fetches all students from the database using Student.query.all().
* get\_student\_by\_id(id): Retrieves a student from the database based on the provided ID using Student.query.get(id).
* create\_student(name, age, email): Creates a new student in the database by creating a Student object, adding it to the session, and committing the changes.
* update\_student(student, name, age, email): Updates an existing student's details by modifying the properties of the provided Student object and committing the changes.
* delete\_student(student): Deletes a student from the database by removing the Student object from the session and committing the changes.

**3. database.py:**

* + This file initializes the SQLAlchemy database instance.
  + The SQLAlchemy class is imported from the flask\_sqlalchemy module.
  + An instance of the SQLAlchemy class is created as db, which will be used to interact with the database.
  + This file serves as a central location to instantiate the database object.

**4. models.py:**

* + This file defines the database model for the Student table.
  + The db instance from the database module is imported.
  + The Student class is defined, representing the student table in the database.
  + The class contains column definitions for id, name, age, and email, which are defined using the db.Column function with the corresponding data types.
  + The \_\_repr\_\_ method is defined to provide a string representation of the Student object.
  + The Student class inherits from db.Model, which is the base class for all models in SQLAlchemy.

**5. schema.gql:**

* + This file defines the GraphQL schema using the GraphQL SDL (Schema Definition Language) syntax.
  + It specifies the types, fields, and relationships for the schema.
  + The Student type is defined with fields id, name, age, and email.
  + The Query type is defined with fields students (returns a list of students) and student (returns a student by ID).
  + The Mutation type is defined with mutation fields createStudent, updateStudent, and deleteStudent for creating, updating, and deleting a student.

**6. schema.py:**

* + This file defines the Strawberry schema using Python classes and decorators.
  + It imports the necessary modules and types (strawberry, db, StudentDAO, List, Union) and defines the schema's types and their fields.
  + The Student class is defined with the @strawberry.type decorator to mark it as a GraphQL type.
  + The Mutation class is defined with mutation methods decorated by @strawberry.mutation.
  + The Query class is defined with query methods decorated by @strawberry.field.
  + A Strawberry schema instance is created with strawberry.Schema(query=Query, mutation=Mutation).

**7. resolvers.py:**

* + This file contains resolver functions for the Strawberry schema.
  + The resolver functions are responsible for executing the logic and fetching the data for each field in the schema.
  + The functions in this file provide the implementation for the query and mutation fields defined in the schema.
  + The functions interact with the DAO (StudentDAO) to fetch, create, update, or delete data from the database.
  + The resolver functions take arguments based on the field parameters defined in the schema and return the corresponding data.

**The app.py sets up the Flask app and configures the database, the dao.py provides methods to interact with the database, the models.py defines the database model, the schema.gql defines the GraphQL schema, the schema.py creates the Strawberry schema, and the resolvers.py provides the resolver functions to execute the GraphQL queries and mutations.**