Final Exam

Source code

```
from flask import Flask
from flask_restx import Api, Resource, fields
from flask_sqlalchemy import SQLAlchemy

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///books.db'
db = SQLAlchemy(app)
api = Api(app)

# Define Book model
book_model = api.model('Book', {
    'ISBN': fields.String,
    'Title': fields.String,
    'Author': fields.String,
    'Genre': fields.String,
    'Price': fields.Float,
    'QuantityAvailable': fields.Integer,
})

# Define Book database model


class Book(db.Model):
    ISBN = db.Column(db.String, primary_key=True)
    Title = db.Column(db.String)
    Author = db.Column(db.String)
    Genre = db.Column(db.String)
    Price = db.Column(db.Float)
    QuantityAvailable = db.Column(db.Integer)


with app.app_context():
    db.create_all()
```

```python
# Define CRUD operations for Books


@api.route('/books')
class BooksResource(Resource):
    @api.marshal_with(book_model, as_list=True)
    def get(self):
        # Get all books
        books = Book.query.all()
        return books

    @api.expect(book_model)
    def post(self):
        # Add a new book
        data = api.payload
        book = Book(**data)
        db.session.add(book)
        db.session.commit()
        return {'message': 'Book added successfully'}, 201


@api.route('/books/<string:isbn>')
class BookResource(Resource):
    @api.marshal_with(book_model)
    def get(self, isbn):
        # Get a book by ISBN
        book = Book.query.get(isbn)
        if book:
            return book
        else:
            return {'message': 'Book not found'}, 404

    def delete(self, isbn):
        # Delete a book by ISBN
        book = Book.query.get(isbn)
        if book:
            db.session.delete(book)
```

```
            db.session.commit()
            return {'message': 'Book deleted successfully'}
        else:
            return {'message': 'Book not found'}, 404

    @api.expect(book_model)
    def put(self, isbn):
        # Update a book by ISBN
        data = api.payload
        book = Book.query.get(isbn)
        if book:
            book.Title = data['Title']
            book.Author = data['Author']
            book.Genre = data['Genre']
            book.Price = data['Price']
            book.QuantityAvailable = data['QuantityAvailable']
            db.session.commit()
            return {'message': 'Book updated successfully'}
        else:
            return {'message': 'Book not found'}, 404


if __name__ == '__main__':
    app.run(debug=True)
```