from enum import unique  
from marshmallow.schema import Schema  
from sqlalchemy.orm import session  
from app import application  
from datetime import datetime  
from flask\_sqlalchemy import SQLAlchemy  
  
""" [DataBase Access Details] Below is the configuration mentioned by which the application can make connection with MySQL database  
"""  
username = 'root'  
password = 'Admin#12345'  
database\_name = 'quiz\_app'  
application.config['SQLALCHEMY\_DATABASE\_URI'] = f"mysql://{username}:{password}@localhost/{database\_name}"  
application.config['SQLALCHEMY\_TRACK\_MODIFICATIONS'] = True  
  
db = SQLAlchemy(application)  
  
  
class UserMaster(db.Model):  
 \_\_tablename\_\_ = 'user\_master'  
  
 id = db.Column(db.String(100), primary\_key=True)  
 name = db.Column(db.String(200), unique=True)  
 username = db.Column(db.String(200), unique=True)  
 password = db.Column(db.String(200))  
 is\_admin = db.Column(db.Integer, default=0)  
 is\_active = db.Column(db.Integer, default=1)  
 created\_ts = db.Column(db.DateTime, default=datetime.utcnow)  
 updated\_ts = db.Column(db.DateTime)  
  
 def \_\_init\_\_(self, id, name, username, password, is\_admin):  
 self.id = id  
 self.name = name  
 self.username = username  
 self.password = password  
 self.is\_admin = is\_admin  
 self.is\_active = 1  
 self.created\_ts = datetime.utcnow()  
  
class UserSession(db.Model):  
 \_\_tablename\_\_ = 'user\_session'  
 id = db.Column(db.String(100), primary\_key=True)  
 user\_id = db.Column(db.String(200), db.ForeignKey("user\_master.id"))  
 session\_id = db.Column(db.String(200), unique=True)  
 is\_active = db.Column(db.Integer, default=1)  
 created\_ts = db.Column(db.DateTime, default=datetime.utcnow)  
 updated\_ts = db.Column(db.DateTime)  
  
 def \_\_init\_\_(self, id, user\_id, session\_id):  
 self.id = id  
 self.user\_id = user\_id  
 self.session\_id = session\_id  
 self.is\_active = 1  
 self.created\_ts = datetime.utcnow()  
  
  
class QuestionMaster(db.Model):  
 \_\_tablename\_\_ = 'question\_master'  
 id = db.Column(db.String(100), primary\_key=True)  
 question = db.Column(db.String(500), unique=True)  
 choice1 = db.Column(db.String(500))  
 choice2 = db.Column(db.String(500))  
 choice3 = db.Column(db.String(500))  
 choice4 = db.Column(db.String(500))  
 answer = db.Column(db.Integer)  
 marks = db.Column(db.Integer)  
 remarks = db.Column(db.String(200))  
 is\_active = db.Column(db.Integer, default=1)  
 created\_ts = db.Column(db.DateTime, default=datetime.utcnow)  
 updated\_ts = db.Column(db.DateTime)  
  
 def \_\_init\_\_(self, id, question, choice1, choice2, choice3, choice4, answer, marks, remarks):  
 self.id = id  
 self.question = question  
 self.choice1 = choice1  
 self.choice2 = choice2  
 self.choice3 = choice3  
 self.choice4 = choice4  
 self.marks = marks  
 self.remarks = remarks  
 self.answer = answer  
 self.is\_active = 1  
 self.created\_ts = datetime.utcnow()  
  
  
class QuizMaster(db.Model):  
 \_\_tablename\_\_ = 'quiz\_master'  
  
 id = db.Column(db.String(100), primary\_key=True)  
 quiz\_name = db.Column(db.String(200), unique=True)  
 is\_active = db.Column(db.Integer, default=1)  
 created\_ts = db.Column(db.DateTime, default=datetime.utcnow)  
 updated\_ts = db.Column(db.DateTime)  
  
 def \_\_init\_\_(self, id, quiz\_name):  
 self.id = id  
 self.quiz\_name = quiz\_name  
 self.is\_active = 1  
 self.created\_ts = datetime.utcnow()  
  
  
class QuizQuestions(db.Model):  
 \_\_tablename\_\_ = 'quiz\_questions'  
 \_\_table\_args\_\_ = (  
 db.UniqueConstraint('quiz\_id', 'question\_id', name='unique\_quiz\_question'),  
 )  
 id = db.Column(db.String(100), primary\_key=True)  
 quiz\_id = db.Column(db.String(200), db.ForeignKey("quiz\_master.id"))  
 question\_id = db.Column(db.String(200), db.ForeignKey("question\_master.id"))  
 is\_active = db.Column(db.Integer, default=1)  
 created\_ts = db.Column(db.DateTime, default=datetime.utcnow)  
 updated\_ts = db.Column(db.DateTime)  
  
 def \_\_init\_\_(self, id, quiz\_id, question\_id):  
 self.id = id  
 self.quiz\_id = quiz\_id  
 self.question\_id = question\_id  
 self.is\_active = 1  
 self.created\_ts = datetime.utcnow()  
  
  
class QuizInstance(db.Model):  
 \_\_tablename\_\_ = 'quiz\_instance'  
 \_\_table\_args\_\_ = (  
 db.UniqueConstraint('quiz\_id', 'user\_id', name='unique\_quiz\_user'),  
 )  
 id = db.Column(db.String(100), primary\_key=True)  
 quiz\_id = db.Column(db.String(200), db.ForeignKey("quiz\_master.id"))  
 user\_id = db.Column(db.String(200), db.ForeignKey("user\_master.id"))  
 score\_achieved = db.Column(db.Integer, default=0)  
 is\_submitted = db.Column(db.Integer, default=0)  
 is\_active = db.Column(db.Integer, default=1)  
 created\_ts = db.Column(db.DateTime, default=datetime.utcnow)  
 updated\_ts = db.Column(db.DateTime)  
  
 def \_\_init\_\_(self, id, quiz\_id, user\_id):  
 self.id = id  
 self.quiz\_id = quiz\_id  
 self.user\_id = user\_id  
 self.score\_achieved = 0  
 self.is\_submitted = 0  
 self.is\_active = 1  
 self.created\_ts = datetime.utcnow()  
  
  
class UserResponses(db.Model):  
 \_\_tablename\_\_ = 'user\_responses'  
 \_\_table\_args\_\_ = (  
 db.UniqueConstraint('quiz\_id', 'user\_id', 'question\_id', name='unique\_quiz\_user\_question'),  
 )  
 id = db.Column(db.String(100), primary\_key=True)  
 quiz\_id = db.Column(db.String(200), db.ForeignKey("quiz\_master.id"))  
 user\_id = db.Column(db.String(200), db.ForeignKey("user\_master.id"))  
 question\_id = db.Column(db.String(200), db.ForeignKey("question\_master.id"))  
 response = db.Column(db.Integer, default=0)  
 is\_active = db.Column(db.Integer, default=1)  
 created\_ts = db.Column(db.DateTime, default=datetime.utcnow)  
 updated\_ts = db.Column(db.DateTime)  
  
 def \_\_init\_\_(self, id, quiz\_id, user\_id, question\_id, response):  
 self.id = id  
 self.quiz\_id = quiz\_id  
 self.user\_id = user\_id  
 self.question\_id = question\_id  
 self.response = response  
 self.is\_active = 1  
 self.created\_ts = datetime.utcnow()  
  
  
db.create\_all()  
db.session.commit()