model classes

```
package com.batr.model;

import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.Column;
import jakarta.persistence.Table;
// import jakarta.persistence.GeneratedValue;
// import jakarta.persistence.GenerationType;

import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.AllArgsConstructor;

@Entity
@Table(name = "category")
@Data // Generates getters, setters, toString, equals, and hashCode
@NoArgsConstructor // No-arg constructor
@AllArgsConstructor // All-arg constructor
public class Category {

    @Id
    // Uncomment if you want the ID to be auto-generated
    // @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    @Column(name = "name", nullable = false)
    private String name;
```

```
    @Column(name = "description")
    private String description;

    @Column(name = "xp_cost")
    private int xp_cost;
}

package com.batr.model;

import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.Column;
import jakarta.persistence.Table;
import jakarta.persistence.Temporal;
import jakarta.persistence.TemporalType;

import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.AllArgsConstructor;

import java.util.Date;

@Entity
@Table(name = "course")
@Data
@NoArgsConstructor
@AllArgsConstructor
```

```
public class Course {

	@Id
	// Uncomment if you want auto-generated ID
	// @GeneratedValue(strategy = GenerationType.IDENTITY)
	private int id;

	@Column(name = "title", nullable = false)
	private String title;

	@Column(name = "description")
	private String description;

	@Column(name = "category_id", nullable = false)
	private String category_id;

	@Column(name = "creator_id", nullable = false)
	private String creator_id;

	@Temporal(TemporalType.TIMESTAMP)
	@Column(name = "created_at")
	private Date created_at;

	@Column(name = "level")
	private String level;
}

package com.batr.model;
```

```
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.Column;
import jakarta.persistence.Table;
import jakarta.persistence.Temporal;
import jakarta.persistence.TemporalType;

import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.AllArgsConstructor;

import java.util.Date;

@Entity
@Table(name = "enrollment")
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Enrollment {

	@Id
	private int id;

	@Column(name = "course_id", nullable = false)
	private int course_id;

	@Column(name = "learner_id", nullable = false)
```

```
    private int learner_id;

    @Temporal(TemporalType.TIMESTAMP)
    @Column(name = "enrollment_date")
    private Date enrollment_date;



}

package com.batr.model;

import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.Column;
import jakarta.persistence.Table;
import jakarta.persistence.Temporal;
import jakarta.persistence.TemporalType;

import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.AllArgsConstructor;

import java.util.Date;

@Entity
@Table(name = "payment")
@Data
```

```
@NoArgsConstructor
@AllArgsConstructor
public class Payment {

	@Id
	private int id;

	@Column(name = "user_id", nullable = false)
	private int user_id;

	@Column(name = "amount", nullable = false)
	private int amount;

	@Column(name = "mode", nullable = false)
	private String mode;

	@Column(name = "xp_purchased")
	private int xp_purchased;

	@Temporal(TemporalType.TIMESTAMP)
	@Column(name = "purchased_at")
	private Date purchased_at;
}

package com.batr.model;

import jakarta.persistence.Entity;
import jakarta.persistence.Id;
```

```
import jakarta.persistence.Column;
import jakarta.persistence.Table;
import jakarta.persistence.Temporal;
import jakarta.persistence.TemporalType;

import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.AllArgsConstructor;

import java.util.Date;

@Entity
@Table(name = "transaction")
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Transaction {

	@Id
	private int id;

	@Column(name = "user_id", nullable = false)
	private int user_id;

	@Column(name = "course_id", nullable = false)
	private int course_id;

	@Column(name = "type", nullable = false)
```

```
    private String type;

    @Column(name = "amount", nullable = false)
    private int amount;

    @Temporal(TemporalType.TIMESTAMP)
    @Column(name = "transacted_at")
    private Date transacted_at;
}

package com.batr.model;

import jakarta.persistence.*;

import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.AllArgsConstructor;

import java.util.Date;
import java.util.List;

@Entity
@Table(name = "user") // Rename if "user" is a reserved word in your DB
@Data
@NoArgsConstructor
@AllArgsConstructor
public class User {
```

```
@Id
@Column(name = "id")
private int id;

@Column(name = "username", nullable = false, unique = true)
private String username;

@Column(name = "email", nullable = false, unique = true)
private String email;

@Column(name = "password", nullable = false)
private String password;

@Column(name = "phone")
private String phone;

@Column(name = "fullname")
private String fullname;

@Column(name = "xp")
private int xp;

@Column(name = "avatar_url")
private String avatar_url;

@Temporal(TemporalType.TIMESTAMP)
@Column(name = "created_at")
private Date created_at;
```

```
    @OneToMany
    @JoinColumn(name = "learner_id", referencedColumnName = "id") // maps to Enrollment.learner_id
    private List<Enrollment> enrollmentList;
}
```

repository classes

```
package com.batr.repository;

import com.batr.model.Category;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface CategoryRepository extends JpaRepository<Category, String> {
  // Add custom query methods if needed
}
package com.batr.repository;

import com.batr.model.Course;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface CourseRepository extends JpaRepository<Course, String> {
  // Add custom query methods if needed
}
```

```
package com.batr.repository;

import com.batr.model.Enrollment;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface EnrollmentRepository extends JpaRepository<Enrollment, String> {
   // Add custom query methods if needed
}

package com.batr.repository;

import com.batr.model.Payment;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface PaymentRepository extends JpaRepository<Payment, String> {
   // Add custom query methods if needed
}
package com.batr.repository;

import com.batr.model.Transaction;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
```

```
@Repository
public interface TransactionRepository extends JpaRepository<Transaction, String> {
   // Add custom query methods if needed
}

package com.batr.repository;

import com.batr.model.User;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import java.util.Optional;

@Repository
public interface UserRepository extends JpaRepository<User, Integer> {
   Optional<User> findByUsername(String username);
   Optional<User> findByEmail(String email);
   Optional<User> findByAuthId(String authId);
}
```

Service interfaces and classes

```
package com.batr.service.impl;

import com.batr.service.CategoryService;

public class CategoryServiceImpl implements CategoryService {
}
```

```
package com.batr.service.impl;

import com.batr.service.CourseService;

public class CourseServiceImpl implements CourseService {
}
package com.batr.service.impl;

import com.batr.service.EnrollmentService;

public class EnrollmentServiceImpl implements EnrollmentService {
}

package com.batr.service.impl;

import com.batr.service.PaymentService;

public class PaymentServiceImpl implements PaymentService {
}

package com.batr.service.impl;

import com.batr.model.User;
import com.batr.repository.UserRepository;
import com.batr.service.UserService;
import lombok.RequiredArgsConstructor;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
```

```
import java.util.Optional;

@RequiredArgsConstructor
@Service
public class UserServiceImpl implements UserService {
  private final UserRepository userRepository;

  public User registerUSer(String email){

    Optional<User> existingUser = userRepository.findByEmail(email);
    if(existingUser.isPresent()){
      throw new RuntimeException("User already registered");
    }

    User user = new User();
    user.setEmail(email);

    return userRepository.save(user);

  }

  @Override
  public Optional<User> getUserByEmail(String email){
    return userRepository.findByEmail(email);
  }

  @Override
```

```
    public void updateXP(int userId, int xpChange){
        User user = userRepository.findById(userId)
                .orElseThrow(() -> new RuntimeException("User not found"));
        user.setXp(user.getXp() + xpChange);
        userRepository.save(user);
    }

    @Override
    public int getUserXP(int userId){
        return userRepository.findById(userId)
                .map(USer::getXp)
                .orElseThrow(() -> new RuntimeException("User not found"));
    }

}
```

```
package com.batr.service;

public interface CategoryService {
}
```

```
package com.batr.service;

public interface CourseService {
}
```

```
package com.batr.service;

public interface EnrollmentService {
```

```
}

package com.batr.service;

public interface PaymentService
{
}
package com.batr.service;

public interface TransactionService {
}



package com.batr.service;

public interface TransactionService {
}
package com.batr.service;

import com.batr.model.User;

import java.util.Optional;

public interface UserService {

   User registerUser(String email);

   Optional<User> getUserByEmail(String email);
```

```
    void updateXP(int userId, int xp);

    int getUserXp(int userId);

}
```

```
package com.batr;

import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest
class BatrApplicationTests {

        @Test
        void contextLoads() {
        }

}
```

```
spring.application.name=batr

# Server port
server.port=8085

# ================================
```

```
# = Data Source Configuration   =
# ================================
spring.datasource.url=jdbc:mysql://localhost:3306/batr
spring.datasource.username=root
spring.datasource.password=98122
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

# ================================
# = JPA Configuration         =
# ================================
spring.jpa.database-platform=org.hibernate.dialect.MySQLDialect
spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto=update
```