User enity

```
package com.crimsonlogic.freelancersystem.entity;

import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.OneToOne;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

import java.util.List;

import com.crimsonlogic.freelancersystem.util.CustomPrefixIdentifierGenerator;

import jakarta.persistence.*;

@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
@Table(name = "users")
public class User {

   @Id
     private String id;  // Custom ID with prefix (e.g., "US")

     private String username;
     private String password;

     @OneToOne(mappedBy = "user")
```

```
    private UserDetails userDetails;

    @OneToMany(mappedBy = "client")
    private List<Project> postedProjects;

    @OneToMany(mappedBy = "freelancer")
    private List<Proposal> submittedProposals;

    @OneToMany(mappedBy = "client")
    private List<Payment> payments;

    @OneToMany(mappedBy = "freelancer")
    private List<Payment> receivedPayments;

    @OneToMany(mappedBy = "client")
    private List<Feedback> givenFeedbacks;

    @OneToMany(mappedBy = "freelancer")
    private List<Feedback> receivedFeedbacks;

    @OneToMany(mappedBy = "user")
    private List<Role> roles;

  @PrePersist
  public void generateId() {
    this.id = CustomPrefixIdentifierGenerator.generateId("US");
  }

}
```

UserDetails entity

```
package com.crimsonlogic.freelancersystem.entity;


import com.crimsonlogic.freelancersystem.util.CustomPrefixIdentifierGenerator;


import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;


@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
@Table(name = "user_details")
public class UserDetails {


  @Id
  private String id;  // This can also have a custom prefix like "UD" for UserDetails


  @Column(nullable = false)
  private String firstName;


  @Column(nullable = false)
  private String lastName;


  @Column(nullable = false)
  private String phoneNumber;


  @Column(nullable = false)
  private String email;

  @OneToOne
  @JoinColumn(name = "user_id", foreignKey = @ForeignKey(name = "FK_USER"))
  private User user;  // The reference back to the User entity

  @PrePersist
  public void generateId() {
    this.id = CustomPrefixIdentifierGenerator.generateId("US");
  }
}
```

Role entity

```
package com.crimsonlogic.freelancersystem.entity;

import com.crimsonlogic.freelancersystem.util.CustomPrefixIdentifierGenerator;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
@Table(name = "roles")
public class Role {

  @Id
```

```
    private String id;

    @Column(nullable = false)
    private String name;

    // Many-to-One relationship with User
    @ManyToOne
    @JoinColumn(name = "user_id", foreignKey = @ForeignKey(name = "FK_USER_ROLE"))
    private User user;

    @PrePersist
    public void generateId() {
        this.id = CustomPrefixIdentifierGenerator.generateId("US");
    }
}
```

ServiceRequest entity

```
package com.crimsonlogic.freelancersystem.entity;

import com.crimsonlogic.freelancersystem.util.CustomPrefixIdentifierGenerator;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
```

```
@Table(name = "service_requests")
public class ServiceRequest {

    @Id
    private String id;  // Custom ID with prefix "SR"

    private String serviceDescription;

    @Column(nullable = false)
    private Double price;

    @Column(nullable = false)
    private String status;

    @ManyToOne
    @JoinColumn(name = "client_id", foreignKey = @ForeignKey(name = "FK_CLIENT_USER"))
    private User client;

    @ManyToOne
    @JoinColumn(name = "freelancer_id", foreignKey = @ForeignKey(name = "FK_FREELANCER_USER"))
    private User freelancer;

    @ManyToOne
    @JoinColumn(name = "service_id", foreignKey = @ForeignKey(name = "FK_SERVICE_REQUEST"))
    private Service service;


    @PrePersist
    public void generateId() {
        this.id = CustomPrefixIdentifierGenerator.generateId("US");
    }

}
```

ServiceCategory entity

```
package com.crimsonlogic.freelancersystem.entity;


import java.util.List;


import com.crimsonlogic.freelancersystem.util.CustomPrefixIdentifierGenerator;


import jakarta.persistence.*;
import jakarta.persistence.PrePersist;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;


@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
@Table(name = "service_categories")
public class ServiceCategory {


    @Id
    private String id;  // Custom ID with prefix "SC"


    @Column(nullable = false)
    private String name;


    @Column(nullable = false)
```

```
  private String description;

  @OneToMany(mappedBy = "serviceCategory")
  private List<Service> services;

  @PrePersist
  public void generateId() {
    this.id = CustomPrefixIdentifierGenerator.generateId("US");
  }
}
```

Service entity

```
package com.crimsonlogic.freelancersystem.entity;

import java.util.List;

import com.crimsonlogic.freelancersystem.util.CustomPrefixIdentifierGenerator;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
@Table(name = "services")
public class Service {
  @Id
  private String id;  // Custom ID with prefix "SRV"

  @Column(nullable = false)
  private String name;

  @Column(nullable = false)
  private String description;

  @Column(nullable = false)
  private Double price;

  @ManyToOne
  @JoinColumn(name = "category_id", foreignKey = @ForeignKey(name = "FK_CATEGORY"))
  private ServiceCategory serviceCategory;

  @ManyToOne
  @JoinColumn(name = "freelancer_id", foreignKey = @ForeignKey(name = "FK_FREELANCER_USER"))
  private User freelancer;

  @PrePersist
  public void generateId() {
    this.id = CustomPrefixIdentifierGenerator.generateId("US");
  }
}
```

Proposal entity

```
package com.crimsonlogic.freelancersystem.entity;

import java.util.List;
import com.crimsonlogic.freelancersystem.util.CustomPrefixIdentifierGenerator;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
@Table(name = "proposals")
public class Proposal {

  @Id
  private String id;  // Custom ID with prefix "PRP"

  @ManyToOne
  @JoinColumn(name = "freelancer_id", foreignKey = @ForeignKey(name = "FK_FREELANCER_USER"))
  private User freelancer;

  @ManyToOne
  @JoinColumn(name = "project_id", foreignKey = @ForeignKey(name = "FK_PROJECT"))
  private Project project;

  @Column(nullable = false)
  private String proposalDetails;

  @Column(nullable = false)
  private Double price;

  @Column(nullable = false)
  private String status;

  @PrePersist
  public void generateId() {
    this.id = CustomPrefixIdentifierGenerator.generateId("US");
  }
}
```

Project entity

```
package com.crimsonlogic.freelancersystem.entity;

import java.util.List;

import com.crimsonlogic.freelancersystem.util.CustomPrefixIdentifierGenerator;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
@Table(name = "projects")
public class Project {
   @Id
  private String id;  // Custom ID with prefix "PR"

   @Column(nullable = false)
  private String name;

   @Column(nullable = false)
  private String description;

   @Column(nullable = false)
   private String status;

   @Column(nullable = false)
    private Double budget;

  @ManyToOne
  @JoinColumn(name = "client_id", foreignKey = @ForeignKey(name = "FK_CLIENT_USER"))
  private User client;

  @OneToMany(mappedBy = "project")
  private List<Proposal> proposals;



  @PrePersist
  public void generateId() {
    this.id = CustomPrefixIdentifierGenerator.generateId("US");
  }
}
```

Payment entity

```
package com.crimsonlogic.freelancersystem.entity;

import com.crimsonlogic.freelancersystem.util.CustomPrefixIdentifierGenerator;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
@Table(name = "payments")
public class Payment {

  @Id
  private String id;  // Custom ID with prefix "TX"

  private Double amount;
  private String paymentStatus;

  @ManyToOne
  @JoinColumn(name = "client_id", foreignKey = @ForeignKey(name = "FK_CLIENT_USER"))
  private User client;

  @ManyToOne
  @JoinColumn(name = "freelancer_id", foreignKey = @ForeignKey(name = "FK_FREELANCER_USER"))
  private User freelancer;
  @ManyToOne
  @JoinColumn(name = "service_request_id", foreignKey = @ForeignKey(name = "FK_SERVICE_REQUEST"))
  private ServiceRequest serviceRequest;

  @PrePersist
  public void generateId() {
    this.id = CustomPrefixIdentifierGenerator.generateId("US");
  }
}
```

Wallet entity

```
package com.crimsonlogic.freelancersystem.entity;

import java.math.BigDecimal;

import com.crimsonlogic.freelancersystem.util.CustomPrefixIdentifierGenerator;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
@Table(name = "wallets")
public class Wallet {

  @Id
  private String id;


  @Column(nullable = false)

  private Double balance;


  // Many-to-One relationship with User

  @ManyToOne

  @JoinColumn(name = "user_id", foreignKey = @ForeignKey(name = "FK_USER_WALLET"))

  private User user;


  @PrePersist

  public void generateId() {

    this.id = CustomPrefixIdentifierGenerator.generateId("US");

  }

  // Constructors, Getters, Setters, etc.

}
```

Feedback entity

```
package com.crimsonlogic.freelancersystem.entity;


import com.crimsonlogic.freelancersystem.util.CustomPrefixIdentifierGenerator;


import jakarta.persistence.*;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.NoArgsConstructor;


@Data

@AllArgsConstructor

@NoArgsConstructor
```

```
@Entity
@Table(name = "feedbacks")
public class Feedback {

    @Id
    private String id;  // Custom ID with prefix "FB"

    private String feedbackText;
    private Integer rating;

    @ManyToOne
    @JoinColumn(name = "client_id", foreignKey = @ForeignKey(name = "FK_CLIENT_USER"))
    private User client;

    @ManyToOne
    @JoinColumn(name = "freelancer_id", foreignKey = @ForeignKey(name = "FK_FREELANCER_USER"))
    private User freelancer;

    @ManyToOne
    @JoinColumn(name = "project_id", foreignKey = @ForeignKey(name = "FK_PROJECT_FEEDBACK"))
    private Project project;

    @PrePersist
    public void generateId() {
        this.id = CustomPrefixIdentifierGenerator.generateId("US");
    }
}
```

User controller

```
package com.crimsonlogic.freelancersystem.controller;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.*;


import com.crimsonlogic.freelancersystem.dto.UserDTO;

import com.crimsonlogic.freelancersystem.service.UserService;


import java.util.List;


@CrossOrigin("*")

@RestController

@RequestMapping("/api/users")

public class UserController {


    @Autowired

    private UserService userService;


    @PostMapping("/register")

    public ResponseEntity<String> createUser(@RequestBody UserDTO userDTO) {

        try {

            userService.createUser(userDTO);

            return ResponseEntity.ok("User registered successfully!");

        } catch (Exception e) {

            return ResponseEntity.badRequest().body("Registration failed: " + e.getMessage());

        }

    }
    @GetMapping("/{id}")
    public ResponseEntity<UserDTO> getUserById(@PathVariable String id) {
        UserDTO userDTO = userService.getUserById(id);
        return ResponseEntity.ok(userDTO);
    }

    @PutMapping("/{id}")
    public ResponseEntity<UserDTO> updateUser(@PathVariable String id, @RequestBody UserDTO userDTO)
{
        UserDTO updatedUser = userService.updateUser(id, userDTO);
        return ResponseEntity.ok(updatedUser);
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<Void> deleteUser(@PathVariable String id) {
        userService.deleteUser(id);
        return ResponseEntity.noContent().build();
    }

    @GetMapping("/getalluser")
    public ResponseEntity<List<UserDTO>> getAllUsers() {
        List<UserDTO> users = userService.getAllUsers();
        return ResponseEntity.ok(users);
    }

    @PostMapping("/login")
    public ResponseEntity<String> login(@RequestBody UserDTO userDTO) {
        UserDTO loggedInUser = userService.login(userDTO);
        return ResponseEntity.ok(loggedInUser.getRole());
    }
}
```

UserDetail controller

```
package com.crimsonlogic.freelancersystem.controller;


import java.util.List;


import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;


import com.crimsonlogic.freelancersystem.dto.UserDetailsDTO;
import com.crimsonlogic.freelancersystem.service.UserDetailsService;


@CrossOrigin("*")
@RestController
@RequestMapping("/api/user-details")
public class UserDetailsController {


  @Autowired
  private UserDetailsService userDetailsService;


  @PostMapping
  public ResponseEntity<UserDetailsDTO> createUserDetails(@RequestBody UserDetailsDTO
userDetailsDTO) {
    UserDetailsDTO createdUserDetails = userDetailsService.createUserDetails(userDetailsDTO);
    return ResponseEntity.ok(createdUserDetails);
  }


  @GetMapping("/{id}")
  public ResponseEntity<UserDetailsDTO> getUserDetailsById(@PathVariable String id) {
    UserDetailsDTO userDetailsDTO = userDetailsService.getUserDetailsById(id);
```

```
        return ResponseEntity.ok(userDetailsDTO);
    }

    @PutMapping("/{id}")
    public ResponseEntity<UserDetailsDTO> updateUserDetails(@PathVariable String id, @RequestBody UserDetailsDTO userDetailsDTO) {
        UserDetailsDTO updatedUserDetails = userDetailsService.updateUserDetails(id, userDetailsDTO);
        return ResponseEntity.ok(updatedUserDetails);
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<Void> deleteUserDetails(@PathVariable String id) {
        userDetailsService.deleteUserDetails(id);
        return ResponseEntity.noContent().build();
    }

    @GetMapping
    public ResponseEntity<List<UserDetailsDTO>> getAllUserDetails() {
        List<UserDetailsDTO> userDetailsList = userDetailsService.getAllUserDetails();
        return ResponseEntity.ok(userDetailsList);
    }
}
```

RoleController

```
package com.crimsonlogic.freelancersystem.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
```

```java
import com.crimsonlogic.freelancersystem.dto.RoleDTO;
import com.crimsonlogic.freelancersystem.service.RoleService;

@CrossOrigin("*")
@RestController
@RequestMapping("/api/roles")
public class RoleController {

    @Autowired
    private RoleService roleService;

    @PostMapping
    public ResponseEntity<RoleDTO> createRole(@RequestBody RoleDTO roleDTO) {
        RoleDTO createdRole = roleService.createRole(roleDTO);
        return ResponseEntity.ok(createdRole);
    }

    @GetMapping("/{id}")
    public ResponseEntity<RoleDTO> getRoleById(@PathVariable String id) {
        RoleDTO roleDTO = roleService.getRoleById(id);
        return ResponseEntity.ok(roleDTO);
    }

    @PutMapping("/{id}")
    public ResponseEntity<RoleDTO> updateRole(@PathVariable String id, @RequestBody RoleDTO roleDTO) {
        RoleDTO updatedRole = roleService.updateRole(id, roleDTO);
        return ResponseEntity.ok(updatedRole);
    }
    @DeleteMapping("/{id}")
    public ResponseEntity<Void> deleteRole(@PathVariable String id) {
        roleService.deleteRole(id);
        return ResponseEntity.noContent().build();
    }

    @GetMapping
    public ResponseEntity<List<RoleDTO>> getAllRoles() {
        List<RoleDTO> roles = roleService.getAllRoles();
        return ResponseEntity.ok(roles);
    }
}
```

ServiceCategoryController

```
package com.crimsonlogic.freelancersystem.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import com.crimsonlogic.freelancersystem.dto.ServiceCategoryDTO;
import com.crimsonlogic.freelancersystem.service.ServiceCategoryService;

import java.util.List;

@CrossOrigin("*")
@RestController
@RequestMapping("/api/service-categories")
public class ServiceCategoryController {
    @Autowired
    private ServiceCategoryService serviceCategoryService;

    @PostMapping
    public ResponseEntity<ServiceCategoryDTO> createServiceCategory(@RequestBody ServiceCategoryDTO serviceCategoryDTO) {
        ServiceCategoryDTO createdCategory = serviceCategoryService.createServiceCategory(serviceCategoryDTO);
        return ResponseEntity.ok(createdCategory);
    }

    @GetMapping("/{id}")
    public ResponseEntity<ServiceCategoryDTO> getServiceCategoryById(@PathVariable String id) {
        ServiceCategoryDTO serviceCategoryDTO = serviceCategoryService.getServiceCategoryById(id);
        return ResponseEntity.ok(serviceCategoryDTO);
    }

    @PutMapping("/{id}")
    public ResponseEntity<ServiceCategoryDTO> updateServiceCategory(@PathVariable String id, @RequestBody ServiceCategoryDTO serviceCategoryDTO) {
        ServiceCategoryDTO updatedCategory = serviceCategoryService.updateServiceCategory(id, serviceCategoryDTO);
        return ResponseEntity.ok(updatedCategory);
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<Void> deleteServiceCategory(@PathVariable String id) {
        serviceCategoryService.deleteServiceCategory(id);
        return ResponseEntity.noContent().build();
    }

    @GetMapping
```

```
  public ResponseEntity<List<ServiceCategoryDTO>> getAllServiceCategories() {
    List<ServiceCategoryDTO> serviceCategories = serviceCategoryService.getAllServiceCategories();
    return ResponseEntity.ok(serviceCategories);
  }
}
```

ServiceRequestController

```
package com.crimsonlogic.freelancersystem.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import com.crimsonlogic.freelancersystem.dto.ServiceRequestDTO;
import com.crimsonlogic.freelancersystem.service.ServiceRequestService;

@CrossOrigin("*")
@RestController
@RequestMapping("/api/service-requests")
public class ServiceRequestController {

  @Autowired
  private ServiceRequestService serviceRequestService;

  @PostMapping
  public ResponseEntity<ServiceRequestDTO> createServiceRequest(@RequestBody ServiceRequestDTO
serviceRequestDTO) {
    ServiceRequestDTO createdServiceRequest =
serviceRequestService.createServiceRequest(serviceRequestDTO);
```

```
        return ResponseEntity.ok(createdServiceRequest);
    }

    @GetMapping("/{id}")
    public ResponseEntity<ServiceRequestDTO> getServiceRequestById(@PathVariable String id) {
        ServiceRequestDTO serviceRequestDTO = serviceRequestService.getServiceRequestById(id);
        return ResponseEntity.ok(serviceRequestDTO);
    }

    @PutMapping("/{id}")
    public ResponseEntity<ServiceRequestDTO> updateServiceRequest(@PathVariable String id,
@RequestBody ServiceRequestDTO serviceRequestDTO) {
        ServiceRequestDTO updatedServiceRequest = serviceRequestService.updateServiceRequest(id,
serviceRequestDTO);
        return ResponseEntity.ok(updatedServiceRequest);
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<Void> deleteServiceRequest(@PathVariable String id) {
        serviceRequestService.deleteServiceRequest(id);
        return ResponseEntity.noContent().build();
    }

    @GetMapping
    public ResponseEntity<List<ServiceRequestDTO>> getAllServiceRequests() {
        List<ServiceRequestDTO> serviceRequests = serviceRequestService.getAllServiceRequests();
        return ResponseEntity.ok(serviceRequests);
    }
}
```

ProposalController

```java
package com.crimsonlogic.freelancersystem.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import com.crimsonlogic.freelancersystem.dto.ProposalDTO;
import com.crimsonlogic.freelancersystem.service.ProposalService;

@CrossOrigin("*")
@RestController
@RequestMapping("/api/proposals")
public class ProposalController {

  @Autowired
  private ProposalService proposalService;

  @PostMapping
  public ResponseEntity<ProposalDTO> createProposal(@RequestBody ProposalDTO proposalDTO) {
    ProposalDTO createdProposal = proposalService.createProposal(proposalDTO);
    return ResponseEntity.ok(createdProposal);
  }

  @GetMapping("/{id}")
  public ResponseEntity<ProposalDTO> getProposalById(@PathVariable String id) {
    ProposalDTO proposalDTO = proposalService.getProposalById(id);
    return ResponseEntity.ok(proposalDTO);
  }
```

```
    @PutMapping("/{id}")
    public ResponseEntity<ProposalDTO> updateProposal(@PathVariable String id, @RequestBody ProposalDTO proposalDTO) {
        ProposalDTO updatedProposal = proposalService.updateProposal(id, proposalDTO);
        return ResponseEntity.ok(updatedProposal);
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<Void> deleteProposal(@PathVariable String id) {
        proposalService.deleteProposal(id);
        return ResponseEntity.noContent().build();
    }

    @GetMapping
    public ResponseEntity<List<ProposalDTO>> getAllProposals() {
        List<ProposalDTO> proposals = proposalService.getAllProposals();
        return ResponseEntity.ok(proposals);
    }
}
```

FeedbackController

```
package com.crimsonlogic.freelancersystem.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import com.crimsonlogic.freelancersystem.dto.FeedbackDTO;
import com.crimsonlogic.freelancersystem.service.FeedbackService;

@CrossOrigin("*")
@RestController
@RequestMapping("/api/feedbacks")
public class FeedbackController {

  @Autowired
  private FeedbackService feedbackService;

  @PostMapping
  public ResponseEntity<FeedbackDTO> createFeedback(@RequestBody FeedbackDTO feedbackDTO) {
    FeedbackDTO createdFeedback = feedbackService.createFeedback(feedbackDTO);
    return ResponseEntity.ok(createdFeedback);
  }

  @GetMapping("/{id}")
  public ResponseEntity<FeedbackDTO> getFeedbackById(@PathVariable String id) {
    FeedbackDTO feedbackDTO = feedbackService.getFeedbackById(id);
    return ResponseEntity.ok(feedbackDTO);
  }

  @PutMapping("/{id}")
  public ResponseEntity<FeedbackDTO> updateFeedback(@PathVariable String id, @RequestBody FeedbackDTO feedbackDTO) {
    FeedbackDTO updatedFeedback = feedbackService.updateFeedback(id, feedbackDTO);
    return ResponseEntity.ok(updatedFeedback);
  }

  @DeleteMapping("/{id}")
  public ResponseEntity<Void> deleteFeedback(@PathVariable String id) {
        feedbackService.deleteFeedback(id);
        return ResponseEntity.noContent().build();
    }

    @GetMapping
    public ResponseEntity<List<FeedbackDTO>> getAllFeedbacks() {
        List<FeedbackDTO> feedbacks = feedbackService.getAllFeedbacks();
        return ResponseEntity.ok(feedbacks);
    }
}
```

ServiceController

```
package com.crimsonlogic.freelancersystem.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import com.crimsonlogic.freelancersystem.dto.ServiceDTO;
import com.crimsonlogic.freelancersystem.service.ServiceService;

import java.util.List;

@CrossOrigin("*")
@RestController
@RequestMapping("/api/services")
public class ServiceController {

    @Autowired
    private ServiceService serviceService;
    @PostMapping
    public ResponseEntity<ServiceDTO> createService(@RequestBody ServiceDTO serviceDTO) {
        ServiceDTO createdService = serviceService.createService(serviceDTO);
        return ResponseEntity.ok(createdService);
    }

    @GetMapping("/{id}")
    public ResponseEntity<ServiceDTO> getServiceById(@PathVariable String id) {
        ServiceDTO serviceDTO = serviceService.getServiceById(id);
        return ResponseEntity.ok(serviceDTO);
    }

    @PutMapping("/{id}")
    public ResponseEntity<ServiceDTO> updateService(@PathVariable String id, @RequestBody ServiceDTO serviceDTO) {
        ServiceDTO updatedService = serviceService.updateService(id, serviceDTO);
        return ResponseEntity.ok(updatedService);
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<Void> deleteService(@PathVariable String id) {
        serviceService.deleteService(id);
        return ResponseEntity.noContent().build();
    }

    @GetMapping
    public ResponseEntity<List<ServiceDTO>> getAllServices() {
        List<ServiceDTO> services = serviceService.getAllServices();
        return ResponseEntity.ok(services);
    }
```

```
}
```

PaymentController

```
package com.crimsonlogic.freelancersystem.controller;


import java.util.List;


import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;


import com.crimsonlogic.freelancersystem.dto.PaymentDTO;
import com.crimsonlogic.freelancersystem.service.PaymentService;


@CrossOrigin("*")
@RestController
@RequestMapping("/api/payments")
public class PaymentController {


  @Autowired
  private PaymentService paymentService;


  @PostMapping
  public ResponseEntity<PaymentDTO> createPayment(@RequestBody PaymentDTO paymentDTO) {
    PaymentDTO createdPayment = paymentService.createPayment(paymentDTO);
    return ResponseEntity.ok(createdPayment);
  }


  @GetMapping("/{id}")
  public ResponseEntity<PaymentDTO> getPaymentById(@PathVariable String id) {
        PaymentDTO paymentDTO = paymentService.getPaymentById(id);
        return ResponseEntity.ok(paymentDTO);
    }

    @PutMapping("/{id}")
    public ResponseEntity<PaymentDTO> updatePayment(@PathVariable String id, @RequestBody PaymentDTO paymentDTO) {
        PaymentDTO updatedPayment = paymentService.updatePayment(id, paymentDTO);
        return ResponseEntity.ok(updatedPayment);
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<Void> deletePayment(@PathVariable String id) {
        paymentService.deletePayment(id);
        return ResponseEntity.noContent().build();
    }

    @GetMapping
    public ResponseEntity<List<PaymentDTO>> getAllPayments() {
        List<PaymentDTO> payments = paymentService.getAllPayments();
        return ResponseEntity.ok(payments);
    }
}
```

ProjectController

```
package com.crimsonlogic.freelancersystem.controller;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import com.crimsonlogic.freelancersystem.dto.ProjectDTO;
```

```java
import com.crimsonlogic.freelancersystem.service.ProjectService;

import java.util.List;

@CrossOrigin("*")
@RestController
@RequestMapping("/api/projects")
public class ProjectController {

    @Autowired
    private ProjectService projectService;

    @PostMapping
    public ResponseEntity<ProjectDTO> createProject(@RequestBody ProjectDTO projectDTO) {
        ProjectDTO createdProject = projectService.createProject(projectDTO);
        return ResponseEntity.ok(createdProject);
    }

    @GetMapping("/{id}")
    public ResponseEntity<ProjectDTO> getProjectById(@PathVariable String id) {
        ProjectDTO projectDTO = projectService.getProjectById(id);
        return ResponseEntity.ok(projectDTO);
    }

    @PutMapping("/{id}")
    public ResponseEntity<ProjectDTO> updateProject(@PathVariable String id, @RequestBody ProjectDTO projectDTO) {
        ProjectDTO updatedProject = projectService.updateProject(id, projectDTO);
        return ResponseEntity.ok(updatedProject);
    }
```

```
  @DeleteMapping("/{id}")

  public ResponseEntity<Void> deleteProject(@PathVariable String id) {

    projectService.deleteProject(id);

    return ResponseEntity.noContent().build();

  }


  @GetMapping

  public ResponseEntity<List<ProjectDTO>> getAllProjects() {

    List<ProjectDTO> projects = projectService.getAllProjects();

    return ResponseEntity.ok(projects);

  }

}
```

WalletController

```
package com.crimsonlogic.freelancersystem.controller;


import java.util.List;


import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.*;


import com.crimsonlogic.freelancersystem.dto.WalletDTO;

import com.crimsonlogic.freelancersystem.service.WalletService;


@CrossOrigin("*")

@RestController

@RequestMapping("/api/wallets")

public class WalletController {
```

```java
    @Autowired
    private WalletService walletService;

    @PostMapping
    public ResponseEntity<WalletDTO> createWallet(@RequestBody WalletDTO walletDTO) {
        WalletDTO createdWallet = walletService.createWallet(walletDTO);
        return ResponseEntity.ok(createdWallet);
    }

    @GetMapping("/{id}")
    public ResponseEntity<WalletDTO> getWalletById(@PathVariable String id) {
        WalletDTO walletDTO = walletService.getWalletById(id);
        return ResponseEntity.ok(walletDTO);
    }

    @PutMapping("/{id}")
    public ResponseEntity<WalletDTO> updateWallet(@PathVariable String id, @RequestBody WalletDTO walletDTO) {
        WalletDTO updatedWallet = walletService.updateWallet(id, walletDTO);
        return ResponseEntity.ok(updatedWallet);
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<Void> deleteWallet(@PathVariable String id) {
        walletService.deleteWallet(id);
        return ResponseEntity.noContent().build();
    }

    @GetMapping
    public ResponseEntity<List<WalletDTO>> getAllWallets() {
        List<WalletDTO> wallets = walletService.getAllWallets();
        return ResponseEntity.ok(wallets);
    }
}
```