编号：

2020 - 2021学年第 二 学期



**实 验 报 告**

实验课程名称 软件架构与应用开发

专 业 班 级 计算机1803

学 生 姓 号 31801150

学 生 姓 名 张帅

实验指导教师 彭彬

一. 实验目的和要求

1.学习并搭建课本8.5的示例（文章管理模块）

2. 利用本节JPA的内容，在实验三的基础上，为学生学生信息管理系统完善数据库设计，并使用JPA完成对基本的学生信息系统中学生基本信息的增删改查询功能。

二. 实验内容

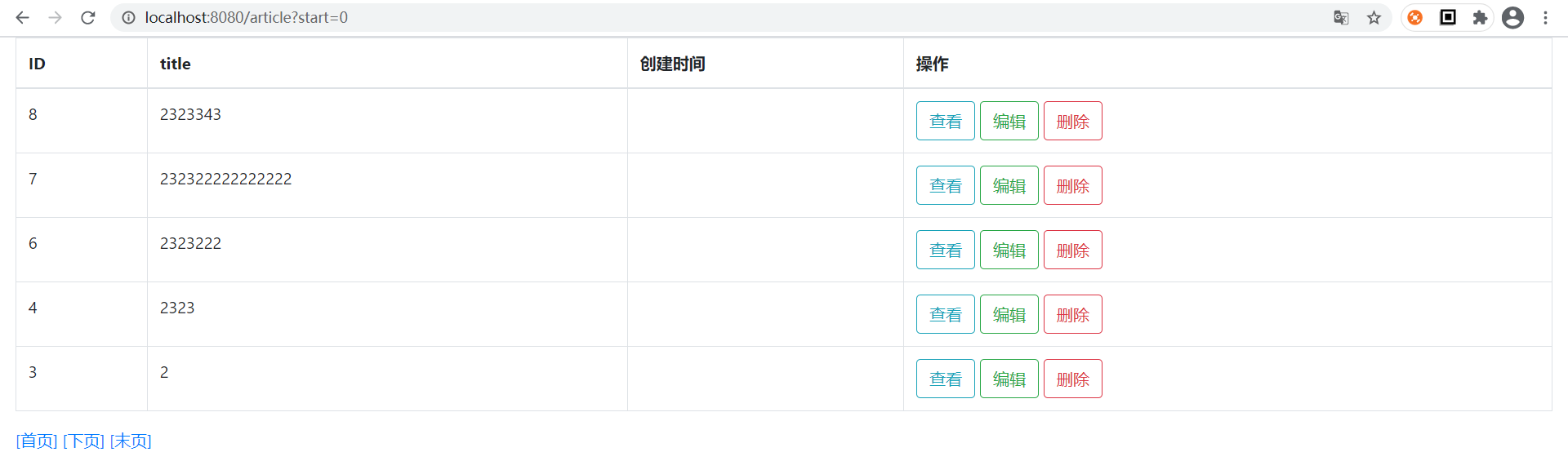
1.学习并搭建课本8.5的示例（文章管理模块）

@Controller  
@RequestMapping("article")  
public class AritcleController {  
 @Autowired  
  
 private ArticleRepository articleRepository;  
  
 @RequestMapping("")  
 public ModelAndView articlelist(@RequestParam(value = "start", defaultValue = "0") Integer start,  
 @RequestParam(value = "limit", defaultValue = "5") Integer limit) {  
 start = start < 0 ? 0 : start;  
 Sort sort = Sort.*by*(Sort.Direction.*DESC*, "id");  
 Pageable pageable = PageRequest.*of*(start, limit, sort);  
 Page<Article> page = articleRepository.findAll(pageable);  
 ModelAndView mav = new ModelAndView("article/list");  
 mav.addObject("page", page);  
 return mav;  
 }  
  
 @GetMapping("/{id}")  
 public ModelAndView getArticle(@PathVariable("id") Integer id) throws Exception {  
 Article articles = articleRepository.findById(id);  
 ModelAndView mav = new ModelAndView("article/show");  
 mav.addObject("article", articles);  
 return mav;  
 }  
  
  
 @GetMapping("/add")  
 public String addArticle() throws Exception{  
 return "article/add";  
 }  
  
  
 @PostMapping("")  
 public String saveArticle(Article model) throws Exception{  
 articleRepository.save(model);  
 return "redirect:/article/";  
 }  
  
 @DeleteMapping("/{id}")  
 public String del(@PathVariable("id") long id) throws Exception{  
 articleRepository.deleteById(id);  
 return "redirect:";  
 }  
  
 @GetMapping("/edit/{id}")  
 public ModelAndView editArticle(@PathVariable("id") long id) throws Exception {  
 Article model = articleRepository.findById(id);  
 ModelAndView mav = new ModelAndView("article/edit");  
 mav.addObject("article", model);  
 return mav;  
 }  
  
 @PutMapping("/{id}")  
 public String editArticleSave(Article model, long id) throws Exception{  
 model.setId(id);  
 articleRepository.save(model);  
 return "redirect:";  
 }  
  
}

@Service  
public class ArticleServiceImpl implements ArticleService {  
 @Autowired  
 private ArticleRepository articleRepository;  
  
  
 @Override  
 public List<Article> getArticleList() {  
 return articleRepository.findAll();  
 }  
  
  
 @Override  
 public Article findArticleById(long id) {  
 return articleRepository.findById(id);  
 }  
  
  
}

public interface ArticleRepository extends JpaRepository<Article,Long> , JpaSpecificationExecutor<Article> {  
 Article findById(long id);  
}

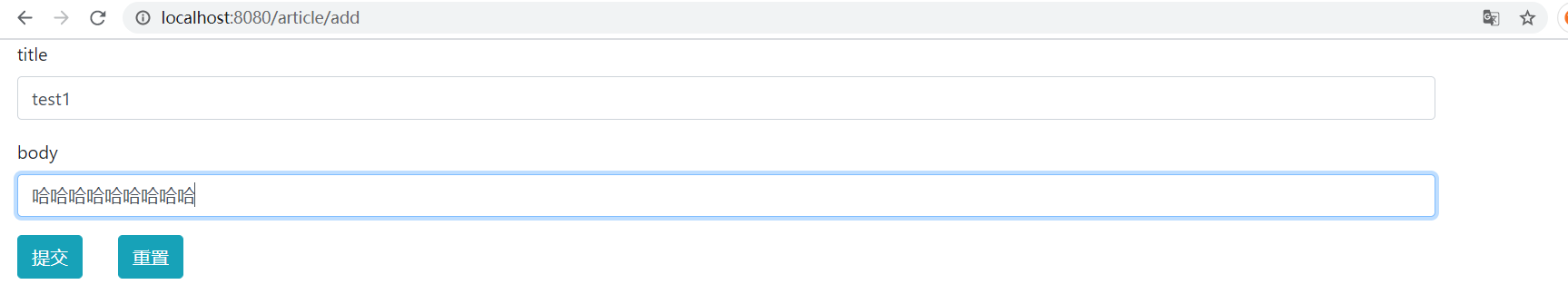
@Entity  
@Data  
public class Article extends BaseEntity implements Serializable {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private long id;  
  
 @Column(nullable = false, unique = true)  
 @NotEmpty(message = "标题不能为空")  
 private String title;  
  
 @Column(nullable = false)  
 private String body;  
  
}

分页查询  


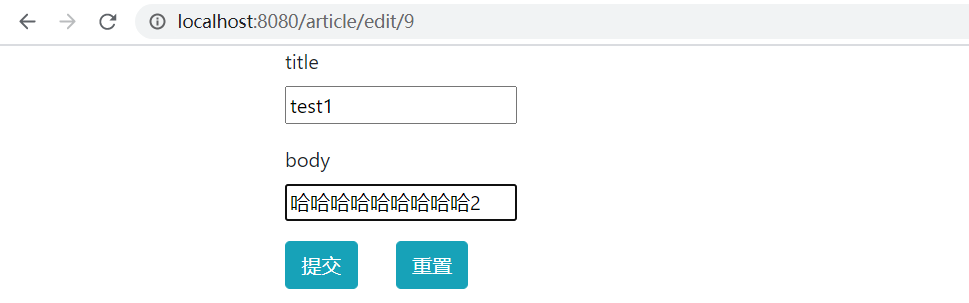


查看一条

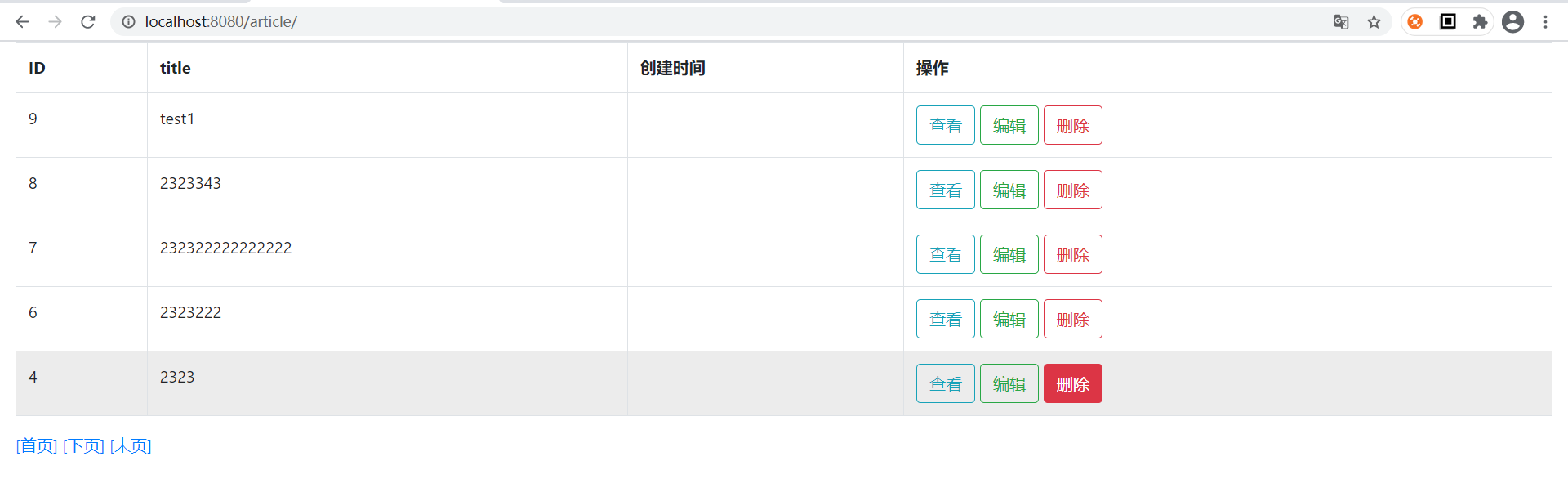


添加  


修改



删除



2. 利用本节JPA的内容，在实验三的基础上，为学生学生信息管理系统完善数据库设计，并使用JPA完成对基本的学生信息系统中学生基本信息的增删改查询功能。

@Entity  
@Data  
public class Stu extends BaseEntity implements Serializable {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private long id;  
  
 @Column(nullable = false, unique = true)  
 @NotNull(message = "学号不能为空")  
 private Integer sid;  
  
 @Column(nullable = false)  
 @NotEmpty(message = "姓名不能为空")  
 @Length(min=1,max=6,message = "姓名长度为1-6")  
 private String name;  
  
 @Column(nullable = false)  
 @NotNull(message = "性别不能为空")  
 @DecimalMax(value = "1",message = "性别异常")  
 @DecimalMin(value = "0",message = "性别异常")  
  
 private Integer sex;  
  
 @Column(nullable = false)  
 @NotNull(message = "年龄不能为空")  
 @DecimalMax(value = "100",message = "年龄太大")  
 @DecimalMin(value = "0",message = "年龄太小")  
 private Integer age;  
  
 @Column(nullable = false)  
 @NotEmpty(message = "邮箱不能为空")  
 @Pattern(regexp = "(^[1-9]\\d{5}(19|([23]\\d))\\d{2}((0[1-9])|(10|11|12))(([0-2][1-9])|10|20|30|31)\\d{3}[0-9Xx]$)|(^[1-9]\\d{5}\\d{2}((0[1-9])|(10|11|12))(([0-2][1-9])|10|20|30|31)\\d{2}[0-9Xx]$)",  
 message = "身份证格式错误")  
 private String email;  
  
 @Column(nullable = false)  
 @Email  
 @NotEmpty(message = "身份证不能为空")  
 private String cardId;  
  
}

public interface StuRepository extends JpaRepository<Stu,Long> , JpaSpecificationExecutor<Stu> {  
 Stu findById(long id);  
}

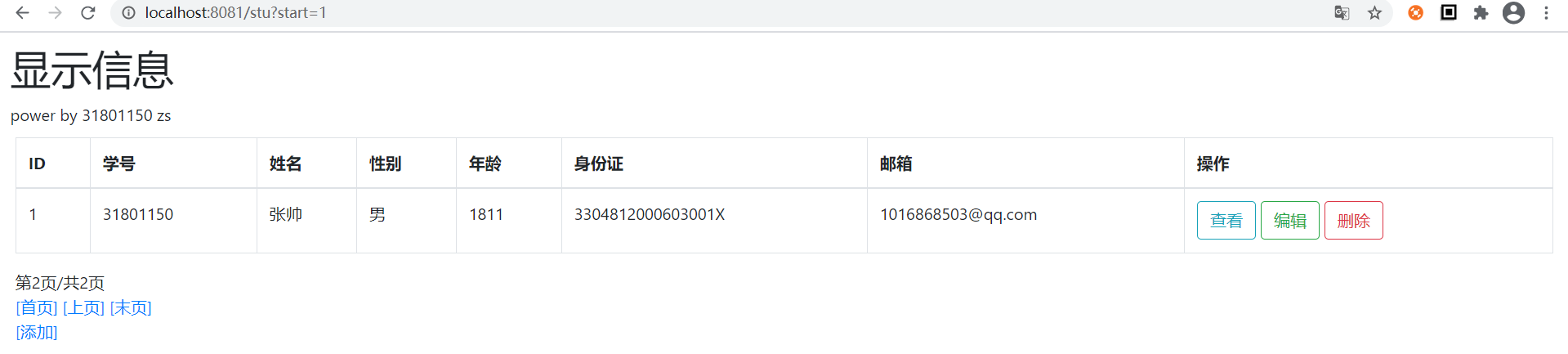
public interface StuService {  
 public List<Stu> getArticleList();  
  
 public Stu findArticleById(long id);  
  
}

@Service  
public class StuServiceImpl implements StuService {  
 @Autowired  
 private StuRepository stuRepository;  
  
  
 @Override  
 public List<Stu> getArticleList() {  
 return stuRepository.findAll();  
 }  
  
  
 @Override  
 public Stu findArticleById(long id) {  
 return stuRepository.findById(id);  
 }  
  
  
}

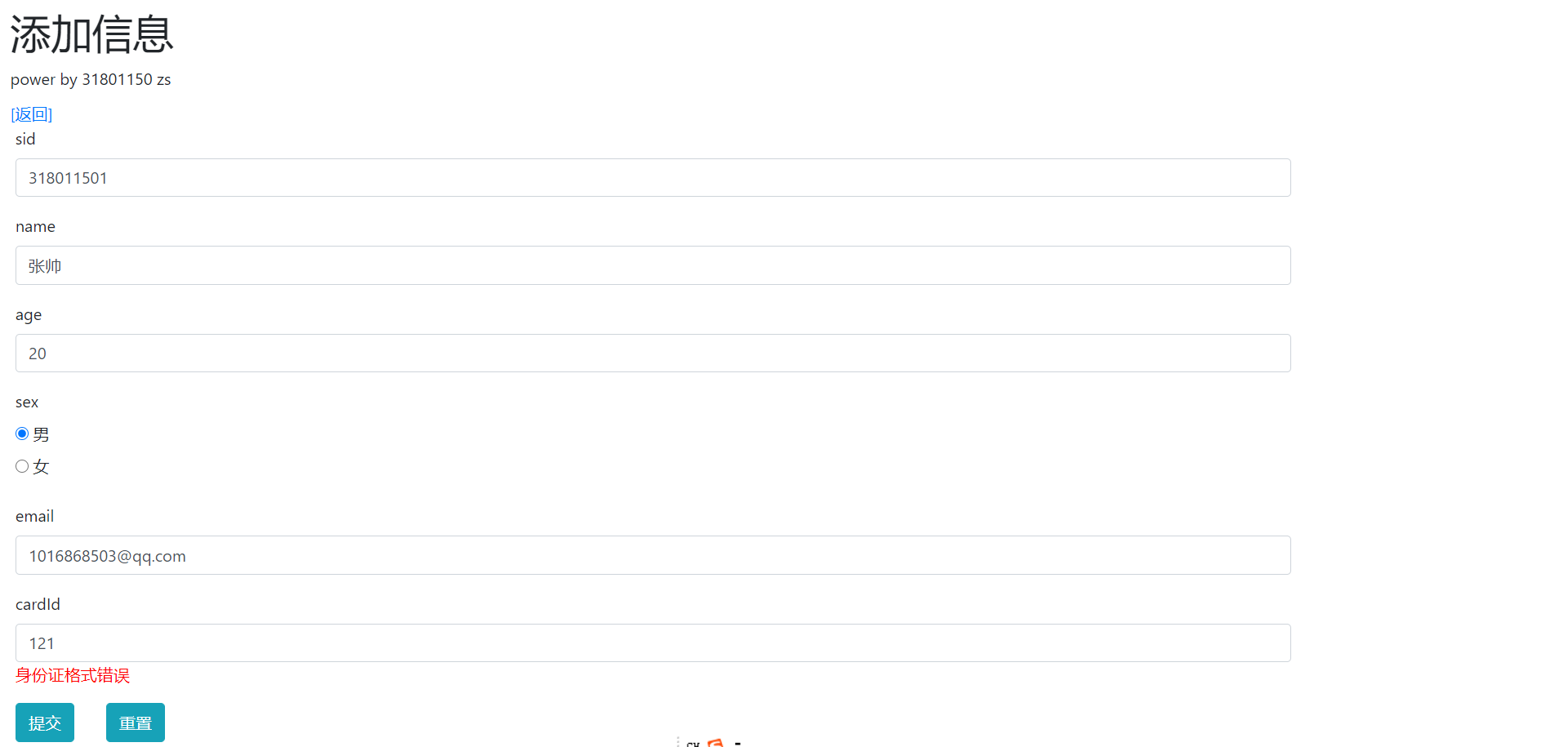
@Controller  
@RequestMapping("stu")  
public class StuController {  
 @Autowired  
  
 private StuRepository stuRepository;  
  
 @RequestMapping("")  
 public ModelAndView Stulist(@RequestParam(value = "start", defaultValue = "0") Integer start,  
 @RequestParam(value = "limit", defaultValue = "5") Integer limit) {  
 start = start < 0 ? 0 : start;  
 Sort sort = Sort.*by*(Sort.Direction.*DESC*, "id");  
 Pageable pageable = PageRequest.*of*(start, limit, sort);  
 Page<Stu> page = stuRepository.findAll(pageable);  
 ModelAndView mav = new ModelAndView("stu/list");  
 mav.addObject("page", page);  
 return mav;  
 }  
  
 @GetMapping("/{id}")  
 public ModelAndView getStu(@PathVariable("id") Integer id) throws Exception {  
 Stu stu = stuRepository.findById(id);  
 ModelAndView mav = new ModelAndView("stu/show");  
 mav.addObject("stu", stu);  
 return mav;  
 }  
  
  
 @GetMapping("/add")  
 public String addStu( Stu model ) throws Exception{  
 return "stu/add";  
 }  
  
  
 @PostMapping("")  
 public String saveStu(@Valid Stu model , BindingResult bindingResult) throws Exception{  
  
  
 if(bindingResult.hasErrors()){  
 return "stu/add";  
 }  
  
 stuRepository.save(model);  
 return "redirect:/stu";  
 }  
  
 @DeleteMapping("/{id}")  
 public String del(@PathVariable("id") long id) throws Exception{  
 stuRepository.deleteById(id);  
 return "redirect:";  
 }  
  
 @GetMapping("/edit/{id}")  
 public ModelAndView editStu(@PathVariable("id") long id) throws Exception {  
 Stu model = stuRepository.findById(id);  
 ModelAndView mav = new ModelAndView("stu/edit");  
 mav.addObject("stu", model);  
 return mav;  
 }  
  
 @PostMapping("/{id}")  
 public String editStuSave(@Valid Stu model,@PathVariable("id") long id) throws Exception{  
  
  
 model.setId(id);  
 stuRepository.save(model);  
 return "redirect:";  
 }  
  
}

分页查找：





添加（验证数据）：

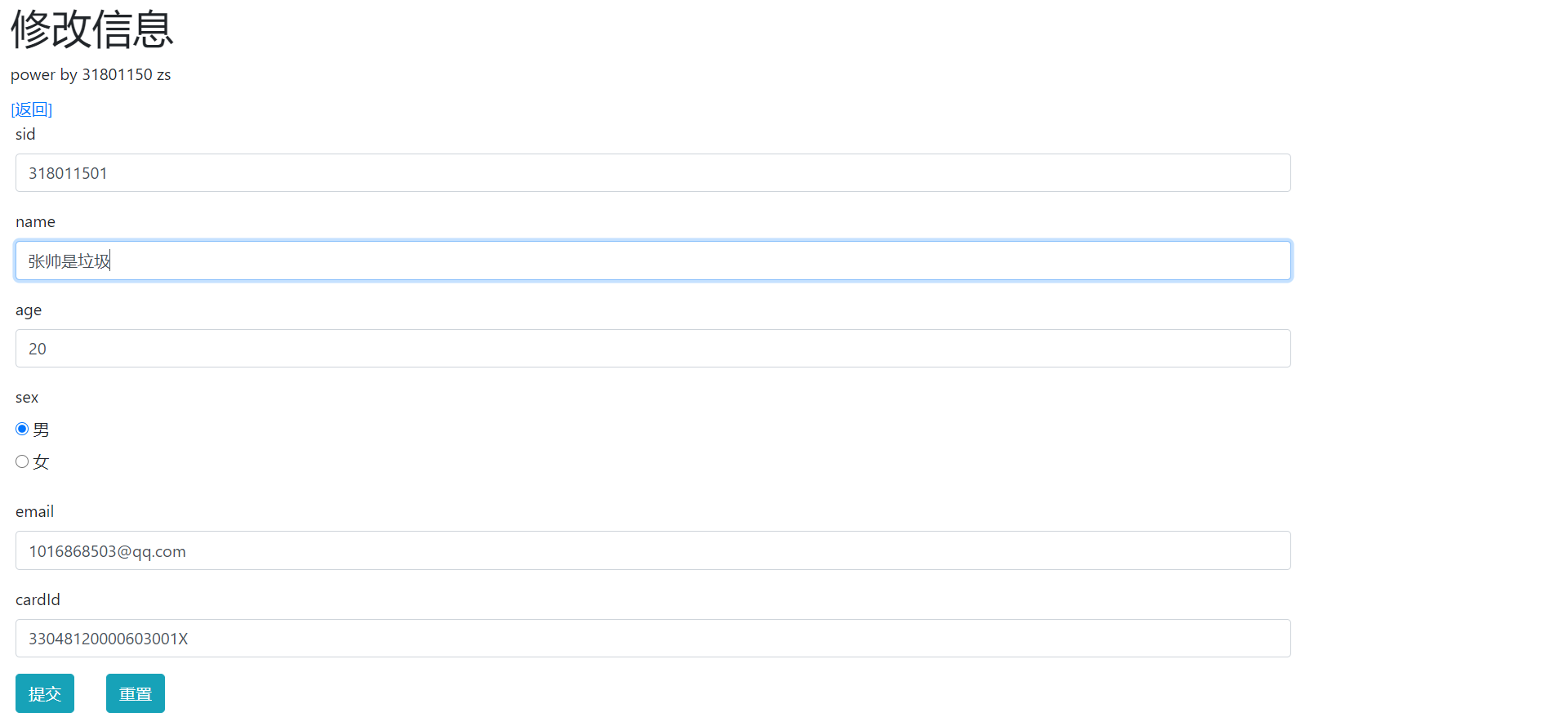


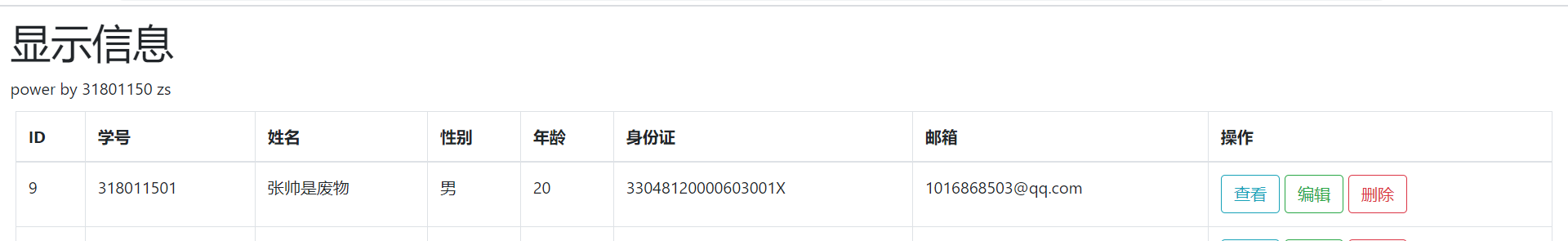


查看数据：



修改数据：





删除数据：  
