|  |  |  |  |
| --- | --- | --- | --- |
| Модули | Описание | Количество строк кода | Размер (в Кбайтах) |
| 1 | 2 | 3 | 4 |
| java\com\hludencov\java\_spring\JavaSpringApplication.java |  | 12 | 1 |
| java\com\hludencov\java\_spring\config\MvcConfig.java |  | 13 | 1 |
| java\com\hludencov\java\_spring\config\WebSecurityConfig.java |  | 55 | 3 |
| java\com\hludencov\java\_spring\controllers\AdminController.java |  | 51 | 2 |
| java\com\hludencov\java\_spring\controllers\AuthController.java |  | 43 | 2 |
| java\com\hludencov\java\_spring\controllers\BlogController.java |  | 111 | 4 |
| java\com\hludencov\java\_spring\controllers\CandidateInfoController.java |  | 105 | 5 |
| java\com\hludencov\java\_spring\controllers\DepartmentController.java |  | 95 | 4 |
| java\com\hludencov\java\_spring\controllers\DocumentController.java |  | 105 | 4 |
| java\com\hludencov\java\_spring\controllers\EducationInstitutionController.java |  | 94 | 4 |
| java\com\hludencov\java\_spring\controllers\GenreController.java |  | 91 | 3 |
| java\com\hludencov\java\_spring\controllers\GroupController.java |  | 102 | 4 |
| java\com\hludencov\java\_spring\controllers\HomeController.java |  | 11 | 1 |
| java\com\hludencov\java\_spring\controllers\MovieController.java |  | 88 | 3 |
| java\com\hludencov\java\_spring\controllers\Personal\_infoController.java |  | 95 | 4 |
| java\com\hludencov\java\_spring\controllers\Preparation\_programController.java |  | 105 | 5 |
| java\com\hludencov\java\_spring\controllers\SubjectController.java |  | 95 | 4 |
| java\com\hludencov\java\_spring\controllers\SummaryController.java |  | 113 | 5 |
| java\com\hludencov\java\_spring\controllers\TeacherInfoController.java |  | 83 | 4 |
| java\com\hludencov\java\_spring\controllers\UsersController.java |  | 107 | 4 |
| java\com\hludencov\java\_spring\models\Candidate\_info.java |  | 75 | 2 |
| java\com\hludencov\java\_spring\models\Department.java |  | 59 | 2 |
| java\com\hludencov\java\_spring\models\Document.java |  | 83 | 2 |
| java\com\hludencov\java\_spring\models\Education\_institution.java |  | 54 | 2 |
| java\com\hludencov\java\_spring\models\Genre.java |  | 48 | 2 |
| java\com\hludencov\java\_spring\models\Group.java |  | 62 | 2 |
| java\com\hludencov\java\_spring\models\Movie.java |  | 96 | 3 |
| java\com\hludencov\java\_spring\models\Personal\_info.java |  | 107 | 3 |
| java\com\hludencov\java\_spring\models\Post.java |  | 77 | 2 |
| java\com\hludencov\java\_spring\models\Preparation\_program.java |  | 63 | 2 |
| java\com\hludencov\java\_spring\models\Role.java |  | 13 | 1 |
| java\com\hludencov\java\_spring\models\Subject.java |  | 96 | 3 |
| java\com\hludencov\java\_spring\models\Summary.java |  | 85 | 2 |
| java\com\hludencov\java\_spring\models\Teacher\_info.java |  | 65 | 2 |
| java\com\hludencov\java\_spring\models\User.java |  | 193 | 5 |
| java\com\hludencov\java\_spring\repo\CandidateRepository.java |  | 10 | 1 |
| java\com\hludencov\java\_spring\repo\DepartmentRepository.java |  | 10 | 1 |
| java\com\hludencov\java\_spring\repo\DocumentRepository.java |  | 12 | 1 |
| java\com\hludencov\java\_spring\repo\Education\_institutionRepository.java |  | 10 | 1 |
| java\com\hludencov\java\_spring\repo\GenreRepository.java |  | 6 | 1 |
| java\com\hludencov\java\_spring\repo\GroupRepository.java |  | 6 | 1 |
| java\com\hludencov\java\_spring\repo\MovieRepository.java |  | 13 | 1 |
| java\com\hludencov\java\_spring\repo\Personal\_infoRepository.java |  | 10 | 1 |
| java\com\hludencov\java\_spring\repo\PostRepository.java |  | 13 | 1 |
| java\com\hludencov\java\_spring\repo\Preparation\_programRepository.java |  | 11 | 1 |
| java\com\hludencov\java\_spring\repo\SubjectRepository.java |  | 11 | 1 |
| java\com\hludencov\java\_spring\repo\SummaryRepository.java |  | 10 | 1 |
| java\com\hludencov\java\_spring\repo\TeachersRepository.java |  | 10 | 1 |
| java\com\hludencov\java\_spring\repo\UserRepository.java |  | 8 | 1 |
| java\com\hludencov\java\_spring\repo\UsersRepository.java |  | 13 | 1 |
| resources\application.properties |  | 9 | 1 |
| resources\templates\blog\blog-add.html |  | 24 | 2 |
| resources\templates\blog\blog-edit.html |  | 26 | 2 |
| resources\templates\blog\blog-filter.html |  | 22 | 1 |
| resources\templates\blog\blog-main.html |  | 28 | 2 |
| resources\templates\blog\blog-show.html |  | 21 | 1 |
| resources\templates\candidate\candidate-add.html |  | 39 | 2 |
| resources\templates\candidate\candidate-edit.html |  | 39 | 2 |
| resources\templates\candidate\candidate-filter.html |  | 32 | 2 |
| resources\templates\candidate\candidate-main.html |  | 34 | 2 |
| resources\templates\candidate\candidate-show.html |  | 16 | 1 |
| resources\templates\department\department-add.html |  | 28 | 2 |
| resources\templates\department\department-edit.html |  | 27 | 2 |
| resources\templates\department\department-filter.html |  | 32 | 2 |
| resources\templates\department\department-main.html |  | 34 | 2 |
| resources\templates\department\department-show.html |  | 15 | 1 |
| resources\templates\document\document-add.html |  | 49 | 3 |
| resources\templates\document\document-edit.html |  | 43 | 3 |
| resources\templates\document\document-filter.html |  | 32 | 2 |
| resources\templates\document\document-main.html |  | 35 | 2 |
| resources\templates\document\document-show.html |  | 18 | 1 |
| resources\templates\education\education-add.html |  | 34 | 2 |
| resources\templates\education\education-edit.html |  | 33 | 2 |
| resources\templates\education\education-filter.html |  | 33 | 2 |
| resources\templates\education\education-main.html |  | 34 | 2 |
| resources\templates\education\education-show.html |  | 16 | 1 |
| resources\templates\genre\genre-add.html |  | 38 | 2 |
| resources\templates\genre\genre-edit.html |  | 39 | 3 |
| resources\templates\genre\genre-main.html |  | 26 | 2 |
| resources\templates\genre\genre-show.html |  | 20 | 1 |
| resources\templates\group\group-add.html |  | 41 | 2 |
| resources\templates\group\group-edit.html |  | 39 | 2 |
| resources\templates\group\group-filter.html |  | 32 | 2 |
| resources\templates\group\group-main.html |  | 32 | 2 |
| resources\templates\group\group-show.html |  | 18 | 1 |
| resources\templates\header.html |  | 195 | 17 |
| resources\templates\home.html |  | 11 | 1 |
| resources\templates\login.html |  | 20 | 1 |
| resources\templates\movie\movie-add.html |  | 56 | 4 |
| resources\templates\movie\movie-edit.html |  | 56 | 4 |
| resources\templates\movie\movie-filter.html |  | 27 | 1 |
| resources\templates\movie\movie-main.html |  | 34 | 2 |
| resources\templates\movie\movie-show.html |  | 20 | 1 |
| resources\templates\personal\_info\personal\_info-add.html |  | 63 | 4 |
| resources\templates\personal\_info\personal\_info-edit.html |  | 62 | 4 |
| resources\templates\personal\_info\personal\_info-filter.html |  | 36 | 2 |
| resources\templates\personal\_info\personal\_info-main.html |  | 35 | 2 |
| resources\templates\personal\_info\personal\_info-show.html |  | 20 | 1 |
| resources\templates\preparation\_program\preparation\_program-add.html |  | 38 | 2 |
| resources\templates\preparation\_program\preparation\_program-edit.html |  | 37 | 2 |
| resources\templates\preparation\_program\preparation\_program-filter.html |  | 32 | 2 |
| resources\templates\preparation\_program\preparation\_program-main.html |  | 34 | 2 |
| resources\templates\preparation\_program\preparation\_program-show.html |  | 15 | 1 |
| resources\templates\registration.html |  | 17 | 1 |
| resources\templates\subject\subject-add.html |  | 34 | 2 |
| resources\templates\subject\subject-edit.html |  | 33 | 2 |
| resources\templates\subject\subject-filter.html |  | 33 | 2 |
| resources\templates\subject\subject-main.html |  | 34 | 2 |
| resources\templates\subject\subject-show.html |  | 16 | 1 |
| resources\templates\summary\summary-add.html |  | 45 | 2 |
| resources\templates\summary\summary-edit.html |  | 37 | 2 |
| resources\templates\summary\summary-filter.html |  | 32 | 2 |
| resources\templates\summary\summary-main.html |  | 35 | 2 |
| resources\templates\summary\summary-show.html |  | 15 | 1 |
| resources\templates\teacher\teacher-add.html |  | 36 | 2 |
| resources\templates\teacher\teacher-edit.html |  | 35 | 2 |
| resources\templates\teacher\teacher-main.html |  | 30 | 2 |
| resources\templates\teacher\teacher-show.html |  | 17 | 1 |
| resources\templates\user\user-add.html |  | 47 | 2 |
| resources\templates\user\user-edit.html |  | 47 | 3 |
| resources\templates\user\user-filter.html |  | 29 | 2 |
| resources\templates\user\user-main.html |  | 33 | 2 |
| resources\templates\user\user-show.html |  | 12 | 1 |

1. java\com\hludencov\java\_spring\JavaSpringApplication.java

package com.hludencov.java\_spring;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class JavaSpringApplication {

public static void main(String[] args) {

SpringApplication.run(JavaSpringApplication.class, args);

}

}

2. java\com\hludencov\java\_spring\config\MvcConfig.java

package com.hludencov.java\_spring.config;

import org.springframework.context.annotation.Configuration;

import org.springframework.web.servlet.config.annotation.ViewControllerRegistry;

import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

@Configuration

public class MvcConfig implements WebMvcConfigurer {

public void addViewControllers(ViewControllerRegistry registry) {

registry.addViewController("/login").setViewName("login");

}

}

3. java\com\hludencov\java\_spring\config\WebSecurityConfig.java

package com.hludencov.java\_spring.config;

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

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;

import org.springframework.security.config.annotation.method.configuration.EnableGlobalMethodSecurity;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;

import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

import org.springframework.security.crypto.password.PasswordEncoder;

import org.springframework.security.web.authentication.logout.SecurityContextLogoutHandler;

import javax.sql.DataSource;

@Configuration

@EnableWebSecurity

@EnableGlobalMethodSecurity(prePostEnabled = true)

public class WebSecurityConfig extends WebSecurityConfigurerAdapter {

@Autowired

private DataSource dataSource;

@Autowired

private PasswordEncoder passwordEncoder;

@Bean

public PasswordEncoder getPasswordEncoder() {

return new BCryptPasswordEncoder(8);

}

@Override

protected void configure(HttpSecurity http) throws Exception {

http

.authorizeRequests()

.antMatchers("/registration").permitAll()

.and()

.formLogin().usernameParameter("login").loginPage("/login").permitAll()

.and()

.logout();

}

@Override

protected void configure(AuthenticationManagerBuilder auth) throws Exception {

auth.jdbcAuthentication()

.dataSource(dataSource)

.passwordEncoder(passwordEncoder)

.usersByUsernameQuery("select login, `password`, active from user where login=?")

.authoritiesByUsernameQuery("select u.login, ur.roles from user u inner join user\_role ur on u.id = ur.user\_id" +

" where u.login=?");

}

}

4. java\com\hludencov\java\_spring\controllers\AdminController.java

package com.hludencov.java\_spring.controllers;

import com.hludencov.java\_spring.models.Role;

import com.hludencov.java\_spring.models.User;

import com.hludencov.java\_spring.repo.UserRepository;

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

import org.springframework.security.access.prepost.PreAuthorize;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

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

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Optional;

@Controller

@RequestMapping("/admin")

@PreAuthorize("hasAnyAuthority('ADMIN')")

public class AdminController {

@Autowired

private UserRepository userRepository;

@GetMapping

public String userList(Model model){

model.addAttribute("users", userRepository.findAll());

return "userList";

}

@GetMapping("/{id}/edit")

public String userEdit(@PathVariable(value = "id") long id, Model model){

Optional<User> user = userRepository.findById(id);

ArrayList<User> res = new ArrayList<>();

user.ifPresent(res::add);

model.addAttribute("user", res);

model.addAttribute("roles", Role.values());

return "userEdit";

}

@PostMapping

public String userSave(@RequestParam String username, @RequestParam(name="roles[]", required = false) String[] roles,

@RequestParam("userId") User user){

user.setLogin(username);

user.getRoles().clear();

if(roles!=null)

{

Arrays.stream(roles).forEach(r->user.getRoles().add(Role.valueOf(r)));

}

userRepository.save(user);

return "redirect:/admin";

}

}

5. java\com\hludencov\java\_spring\controllers\AuthController.java

package com.hludencov.java\_spring.controllers;

import com.hludencov.java\_spring.models.Role;

import com.hludencov.java\_spring.models.User;

import com.hludencov.java\_spring.repo.UserRepository;

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

import org.springframework.security.crypto.password.PasswordEncoder;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

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

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

import java.util.Collections;

@Controller

public class AuthController {

@Autowired

private UserRepository userRepository;

@Autowired

private PasswordEncoder passwordEncoder;

@GetMapping("/registration")

public String registration(){

return "registration";

}

@PostMapping("/registration")

public String addUser(User user, Model model){

User userFromDb = userRepository.findByLogin(user.getLogin());

if(userFromDb != null)

{

model.addAttribute("message", "Пользователь с таким логином уже зарегистрирован");

return "registration";

}

user.setActive(true);

user.setRoles(Collections.singleton(Role.USER));

user.setPassword(passwordEncoder.encode(user.getPassword()));

userRepository.save(user);

return "redirect:/login";

}

}

6. java\com\hludencov\java\_spring\controllers\BlogController.java

package com.hludencov.java\_spring.controllers;

import com.hludencov.java\_spring.models.Movie;

import com.hludencov.java\_spring.models.User;

import com.hludencov.java\_spring.repo.MovieRepository;

import com.hludencov.java\_spring.repo.PostRepository;

import com.hludencov.java\_spring.repo.UsersRepository;

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

import com.hludencov.java\_spring.models.Post;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

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

import javax.validation.Valid;

import java.util.List;

@Controller

public class BlogController {

@Autowired

PostRepository postRepository;

@Autowired

MovieRepository movieRepository;

@Autowired

UsersRepository usersRepository;

@GetMapping("/blog")

public String blogMain(Model model) {

Iterable<Post> posts = postRepository.findAll();

model.addAttribute("posts", posts);

return "blog/blog-main";

}

@GetMapping("/blog/add")

public String blogAdd(Model model, Post post) {

Iterable<User> users = usersRepository.findAll();

Iterable<Movie> movie = movieRepository.findAll();

model.addAttribute("users", users);

model.addAttribute("movie", movie);

return "blog/blog-add";

}

@PostMapping("/blog/add")

public String moviePostAdd(@ModelAttribute("post") @Valid Post post, BindingResult bindingResult) {

if (bindingResult.hasErrors()) {

return "blog/blog-add";

}

postRepository.save(post);

return "redirect:/blog";

}

@GetMapping("blog/edit/{post}")

public String postEdit(Post post, Model model) {

Iterable<User> users = usersRepository.findAll();

Iterable<Movie> movie = movieRepository.findAll();

model.addAttribute("users", users);

model.addAttribute("movie", movie);

return "blog/blog-edit";

}

@PostMapping("blog/edit/{post}")

public String postPostEdit(

@ModelAttribute("post") @Valid Post post,

BindingResult bindingResult,

Model model

) {

if (bindingResult.hasErrors()) {

Iterable<User> users = usersRepository.findAll();

Iterable<Movie> movie = movieRepository.findAll();

model.addAttribute("users", users);

model.addAttribute("movie", movie);

return "blog/blog-edit";

}

postRepository.save(post);

return "redirect:../";

}

@GetMapping("/blog/filter")

public String blogFilter(Model model) {

return "blog/blog-filter";

}

@PostMapping("/blog/filter/result")

public String blogResult(@RequestParam String title, Model model) {

List<Post> result = postRepository.findByTitleContains(title);

// List<Post> result = postRepository.findLikeTitle(title);

model.addAttribute("result", result);

return "blog/blog-filter";

}

@GetMapping("/blog/show/{post}")

public String postShow(

Post post,

Model model) {

model.addAttribute("post", post);

return "blog/blog-show";

}

@GetMapping("/del/{post}")

public String postDel(

Post post) {

postRepository.delete(post);

return "redirect:../";

}

}

7. java\com\hludencov\java\_spring\controllers\CandidateInfoController.java

package com.hludencov.java\_spring.controllers;

import com.hludencov.java\_spring.models.Candidate\_info;

import com.hludencov.java\_spring.repo.CandidateRepository;

import com.hludencov.java\_spring.repo.DepartmentRepository;

import com.hludencov.java\_spring.repo.UserRepository;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

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

import javax.validation.Valid;

import java.util.List;

@Controller

@RequestMapping("/candidate")

public class CandidateInfoController {

@Autowired

private CandidateRepository candidateRepository;

@Autowired

private DepartmentRepository departmentRepository;

@Autowired

private UserRepository userRepository;

@GetMapping

public String candidateList(Model model) {

model.addAttribute("candidates", candidateRepository.findAll());

return "candidate/candidate-main";

}

@GetMapping("/add")

public String candidateAdd(Candidate\_info candidate, Model model) {

model.addAttribute("candidates", candidateRepository.findAll());

model.addAttribute("departments", departmentRepository.findAll());

model.addAttribute("users", userRepository.findAll());

return "candidate/candidate-add";

}

@PostMapping("/add")

public String candidatePostAdd(@ModelAttribute("candidate\_info") @Valid Candidate\_info candidate, BindingResult bindingResult, Model model) {

if (bindingResult.hasErrors()) {

model.addAttribute("candidates", candidateRepository.findAll());

model.addAttribute("departments", departmentRepository.findAll());

model.addAttribute("users", userRepository.findAll());

return "candidate/candidate-add";

}

candidateRepository.save(candidate);

return "redirect:/candidate";

}

@GetMapping("/edit/{candidate\_info}")

public String candidateEdit(

Candidate\_info candidate,

Model model) {

model.addAttribute("departments", departmentRepository.findAll());

model.addAttribute("users", userRepository.findAll());

model.addAttribute("candidates", candidate);

return "candidate/candidate-edit";

}

@PostMapping("/edit/{candidate\_info}")

public String candidatePostEdit(@ModelAttribute("candidate\_info") @Valid Candidate\_info candidate, BindingResult bindingResult, Model model) {

if (bindingResult.hasErrors()) {

model.addAttribute("candidates", candidateRepository.findAll());

model.addAttribute("departments", departmentRepository.findAll());

model.addAttribute("users", userRepository.findAll());

return "candidate/candidate-edit";

}

candidateRepository.save(candidate);

return "redirect:../";

}

@GetMapping("/show/{candidate\_info}")

public String candidateShow(

Candidate\_info candidate,

Model model) {

model.addAttribute("departments", departmentRepository.findAll());

model.addAttribute("users", userRepository.findAll());

model.addAttribute("candidates", candidate);

return "candidate/candidate-show";

}

@GetMapping("/del/{candidate\_info}")

public String candidateDel(

Candidate\_info candidate) {

candidateRepository.delete(candidate);

return "redirect:../";

}

@GetMapping("/filter")

public String candidateFilter(Model model) {

return "candidate/candidate-filter";

}

@PostMapping("/filter/result")

public String candidateResult(@RequestParam String title, Model model) {

List<Candidate\_info> result = candidateRepository.findBySubmissionDateContains(title);

model.addAttribute("result", result);

return "candidate/candidate-filter";

}

}

8. java\com\hludencov\java\_spring\controllers\DepartmentController.java

package com.hludencov.java\_spring.controllers;

import com.hludencov.java\_spring.models.Department;

import com.hludencov.java\_spring.repo.DepartmentRepository;

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

import org.springframework.security.access.prepost.PreAuthorize;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

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

import javax.validation.Valid;

import java.util.List;

@Controller

@RequestMapping("/department")

@PreAuthorize("hasAnyAuthority('ADMIN', 'TEACHER')")

public class DepartmentController {

@Autowired

private DepartmentRepository departmentRepository;

@GetMapping

public String educationList(Department department, Model model){

model.addAttribute("department", departmentRepository.findAll());

return "department/department-main";

}

@GetMapping("/add")

public String educationAdd(Department department) {

return "department/department-add";

}

@PostMapping("/add")

public String educationPostAdd(@ModelAttribute("department") @Valid Department department, BindingResult bindingResult) {

if (bindingResult.hasErrors()) {

return "department/department-add";

}

departmentRepository.save(department);

return "redirect:/department";

}

@GetMapping("/edit/{department}")

public String educationEdit(

Department department,

Model model) {

model.addAttribute("department", department);

return "department/department-edit";

}

@PostMapping("/edit/{department}")

public String educationPostEdit(@ModelAttribute("department") @Valid Department department, BindingResult bindingResult) {

if (bindingResult.hasErrors()) {

return "department/department-edit";

}

departmentRepository.save(department);

return "redirect:../";

}

@GetMapping("/show/{department}")

public String educationShow(

Department department,

Model model) {

model.addAttribute("department", department);

return "department/department-show";

}

@GetMapping("/del/{department}")

public String educationDel(

Department department) {

departmentRepository.delete(department);

return "redirect:../";

}

@GetMapping("/filter")

public String educationFilter(Model model) {

return "department/department-filter";

}

@PostMapping("/filter/result")

public String educationResult(@RequestParam String title, Model model) {

List<Department> result = departmentRepository.findByNameContains(title);

model.addAttribute("result", result);

return "department/department-filter";

}

@PostMapping("/filter\_strict/result")

public String educationStrictResult(@RequestParam String title, Model model) {

List<Department> result = departmentRepository.findByName(title);

model.addAttribute("result", result);

return "department/department-filter";

}

}

9. java\com\hludencov\java\_spring\controllers\DocumentController.java

package com.hludencov.java\_spring.controllers;

import com.hludencov.java\_spring.models.Document;

import com.hludencov.java\_spring.repo.DocumentRepository;

import com.hludencov.java\_spring.repo.UserRepository;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

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

import javax.validation.Valid;

import java.util.List;

@Controller

@RequestMapping("/document")

public class DocumentController {

@Autowired

private DocumentRepository documentRepository;

@Autowired

private UserRepository userRepository;

@GetMapping

public String documentList(Model model) {

model.addAttribute("documents", documentRepository.findAll());

return "document/document-main";

}

@GetMapping("/add")

public String documentAdd(Document document, Model model) {

model.addAttribute("documents", documentRepository.findAll());

model.addAttribute("users", userRepository.findAll());

return "document/document-add";

}

@PostMapping("/add")

public String documentPostAdd(@ModelAttribute("document") @Valid Document document, BindingResult bindingResult, Model model) {

if (bindingResult.hasErrors()) {

model.addAttribute("documents", documentRepository.findAll());

model.addAttribute("users", userRepository.findAll());

return "document/document-add";

}

documentRepository.save(document);

return "redirect:/document";

}

@GetMapping("/edit/{document}")

public String documentEdit(

Document document,

Model model) {

model.addAttribute("users", userRepository.findAll());

model.addAttribute("documents", document);

return "document/document-edit";

}

@PostMapping("/edit/{document}")

public String documentPostEdit(@ModelAttribute("document") @Valid Document document, BindingResult bindingResult, Model model) {

if (bindingResult.hasErrors()) {

model.addAttribute("documents", documentRepository.findAll());

model.addAttribute("users", userRepository.findAll());

return "document/document-edit";

}

documentRepository.save(document);

return "redirect:../";

}

@GetMapping("/show/{document}")

public String documentShow(

Document document,

Model model) {

model.addAttribute("users", userRepository.findAll());

model.addAttribute("documents", document);

return "document/document-show";

}

@GetMapping("/del/{document}")

public String documentDel(

Document document) {

documentRepository.delete(document);

return "redirect:../";

}

@GetMapping("/filter")

public String documentFilter(Model model) {

return "document/document-filter";

}

@PostMapping("/filter/result")

public String documentResult(@RequestParam String title, Model model) {

List<Document> result = documentRepository.findByFileNameContains(title);

model.addAttribute("result", result);

return "document/document-filter";

}

@PostMapping("/filter\_strict/result")

public String documentStrictResult(@RequestParam String title, Model model) {

List<Document> result = documentRepository.findByFileName(title);

model.addAttribute("result", result);

return "document/document-filter";

}

}

10. java\com\hludencov\java\_spring\controllers\EducationInstitutionController.java

package com.hludencov.java\_spring.controllers;

import com.hludencov.java\_spring.models.Education\_institution;

import com.hludencov.java\_spring.repo.Education\_institutionRepository;

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

import org.springframework.security.access.prepost.PreAuthorize;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

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

import javax.validation.Valid;

import java.util.List;

@Controller

@RequestMapping("/education")

@PreAuthorize("hasAnyAuthority('ADMIN', 'TEACHER')")

public class EducationInstitutionController {

@Autowired

private Education\_institutionRepository education\_institutionRepository;

@GetMapping

public String educationList(Education\_institution education\_institution,Model model){

model.addAttribute("education\_institution", education\_institutionRepository.findAll());

return "education/education-main";

}

@GetMapping("/add")

public String educationAdd(Education\_institution education\_institution) {

return "education/education-add";

}

@PostMapping("/add")

public String educationPostAdd(@ModelAttribute("education\_institution") @Valid Education\_institution education\_institution, BindingResult bindingResult) {

if (bindingResult.hasErrors()) {

return "education/education-add";

}

education\_institutionRepository.save(education\_institution);

return "redirect:/education";

}

@GetMapping("/edit/{education\_institution}")

public String educationEdit(

Education\_institution education\_institution,

Model model) {

model.addAttribute("education\_institution", education\_institution);

return "education/education-edit";

}

@PostMapping("/edit/{education\_institution}")

public String educationPostEdit(@ModelAttribute("education\_institution") @Valid Education\_institution education\_institution, BindingResult bindingResult) {

if (bindingResult.hasErrors()) {

return "education/education-edit";

}

education\_institutionRepository.save(education\_institution);

return "redirect:../";

}

@GetMapping("/show/{education\_institution}")

public String educationShow(

Education\_institution education\_institution,

Model model) {

model.addAttribute("education\_institution", education\_institution);

return "education/education-show";

}

@GetMapping("/del/{education\_institution}")

public String educationDel(

Education\_institution education\_institution) {

education\_institutionRepository.delete(education\_institution);

return "redirect:../";

}

@GetMapping("/filter")

public String educationFilter(Model model) {

return "education/education-filter";

}

@PostMapping("/filter/result")

public String educationResult(@RequestParam String title, Model model) {

List<Education\_institution> result = education\_institutionRepository.findByNameContains(title);

model.addAttribute("result", result);

return "education/education-filter";

}

@PostMapping("/filter\_strict/result")

public String educationStrictResult(@RequestParam String title, Model model) {

List<Education\_institution> result = education\_institutionRepository.findByName(title);

model.addAttribute("result", result);

return "education/education-filter";

}

}

11. java\com\hludencov\java\_spring\controllers\GenreController.java

package com.hludencov.java\_spring.controllers;

import com.hludencov.java\_spring.models.Genre;

import com.hludencov.java\_spring.models.Movie;

import com.hludencov.java\_spring.repo.GenreRepository;

import com.hludencov.java\_spring.repo.MovieRepository;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

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

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

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

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

import javax.validation.Valid;

@Controller

@RequestMapping("genre")

public class GenreController {

@Autowired

private GenreRepository genreRepository;

@Autowired

private MovieRepository movieRepository;

@GetMapping()

public String genreMain(Model model) {

Iterable<Genre> genres = genreRepository.findAll();

model.addAttribute("genres", genres);

return "genre/genre-main";

}

@GetMapping("/add")

public String genreAdd(Genre genre, Model model) {

Iterable<Movie> movie = movieRepository.findAll();

model.addAttribute("movie", movie);

return "genre/genre-add";

}

@PostMapping("/add")

public String genrePostAdd(

@ModelAttribute("genre") @Valid Genre genre,

BindingResult bindingResult,

Model model

) {

if (bindingResult.hasErrors()) {

Iterable<Movie> movie = movieRepository.findAll();

model.addAttribute("movie", movie);

return "genre/genre-add";

}

genreRepository.save(genre);

return "redirect:";

}

@GetMapping("/edit/{genre}")

public String genreEdit(Genre genre, Model model) {

Iterable<Movie> movie = movieRepository.findAll();

model.addAttribute("movie", movie);

return "genre/genre-edit";

}

@PostMapping("/edit/{genre}")

public String genrePostEdit(

@ModelAttribute("genre") @Valid Genre genre,

BindingResult bindingResult,

Model model

) {

if (bindingResult.hasErrors()) {

Iterable<Movie> movie = movieRepository.findAll();

model.addAttribute("movie", movie);

return "genre/genre-edit";

}

genreRepository.save(genre);

return "redirect:../";

}

@GetMapping("/show/{genre}")

public String genreShow(

Genre genre, Model model) {

model.addAttribute("genre", genre);

return "genre/genre-show";

}

@GetMapping("/del/{genre}")

public String genreDel(

Genre genre) {

genreRepository.delete(genre);

return "redirect:../";

}

}

12. java\com\hludencov\java\_spring\controllers\GroupController.java

package com.hludencov.java\_spring.controllers;

import com.hludencov.java\_spring.models.Group;

import com.hludencov.java\_spring.repo.GroupRepository;

import com.hludencov.java\_spring.repo.Preparation\_programRepository;

import com.hludencov.java\_spring.repo.UserRepository;

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

import org.springframework.security.access.prepost.PreAuthorize;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

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

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

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

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

import javax.validation.Valid;

@Controller

@RequestMapping("/group")

@PreAuthorize("hasAnyAuthority('ADMIN')")

public class GroupController {

@Autowired

private GroupRepository groupRepository;

@Autowired

private UserRepository userRepository;

@Autowired

private Preparation\_programRepository preparation\_programRepository;

@GetMapping

public String educationList(Group group, Model model) {

model.addAttribute("group", groupRepository.findAll());

return "group/group-main";

}

@GetMapping("/add")

public String educationAdd(Group group, Model model) {

model.addAttribute("users", userRepository.findAll());

model.addAttribute("preparation\_programs", preparation\_programRepository.findAll());

return "group/group-add";

}

@PostMapping("/add")

public String educationPostAdd(@ModelAttribute("group") @Valid Group group, BindingResult bindingResult, Model model) {

if (bindingResult.hasErrors()) {

model.addAttribute("users", userRepository.findAll());

model.addAttribute("preparation\_programs", preparation\_programRepository.findAll());

return "group/group-add";

}

groupRepository.save(group);

return "redirect:/group";

}

@GetMapping("/edit/{group}")

public String educationEdit(

Group group,

Model model) {

model.addAttribute("users", userRepository.findAll());

model.addAttribute("preparation\_programs", preparation\_programRepository.findAll());

model.addAttribute("group", group);

return "group/group-edit";

}

@PostMapping("/edit/{group}")

public String educationPostEdit(@ModelAttribute("group") @Valid Group group, BindingResult bindingResult, Model model) {

if (bindingResult.hasErrors()) {

model.addAttribute("users", userRepository.findAll());

model.addAttribute("preparation\_programs", preparation\_programRepository.findAll());

return "group/group-edit";

}

groupRepository.save(group);

return "redirect:../";

}

@GetMapping("/show/{group}")

public String educationShow(

Group group,

Model model) {

model.addAttribute("users", userRepository.findAll());

model.addAttribute("preparation\_programs", preparation\_programRepository.findAll());

model.addAttribute("groups", group);

return "group/group-show";

}

@GetMapping("/del/{group}")

public String educationDel(

Group group) {

groupRepository.delete(group);

return "redirect:../";

}

@GetMapping("/filter")

public String educationFilter(Model model) {

return "group/group-filter";

}

}

13. java\com\hludencov\java\_spring\controllers\HomeController.java

package com.hludencov.java\_spring.controllers;

import org.springframework.stereotype.Controller;

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

@Controller

public class HomeController {

@GetMapping()

public String bookMain() {

return "home";

}

}

14. java\com\hludencov\java\_spring\controllers\MovieController.java

package com.hludencov.java\_spring.controllers;

import com.hludencov.java\_spring.models.Movie;

import com.hludencov.java\_spring.repo.MovieRepository;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

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

import javax.validation.Valid;

import java.util.List;

@Controller

public class MovieController {

@Autowired

MovieRepository movieRepository;

@GetMapping("/movie")

public String movieMain(Model model) {

Iterable<Movie> movie = movieRepository.findAll();

model.addAttribute("movie", movie);

return "movie/movie-main";

}

@GetMapping("/movie/add")

public String movieAdd(Movie movie) {

return "movie/movie-add";

}

@PostMapping("/movie/add")

public String moviePostAdd(@ModelAttribute("movie") @Valid Movie movie, BindingResult bindingResult) {

if(bindingResult.hasErrors()){

return "movie/movie-add";

}

movieRepository.save(movie);

return "redirect:/movie";

}

@GetMapping("/movie/edit/{movie}")

public String movieEdit(

Movie movie,

Model model) {

model.addAttribute("movie", movie);

return "movie/movie-edit";

}

@PostMapping("/movie/edit/{movie}")

public String moviePostEdit(@ModelAttribute("movie") @Valid Movie movie, BindingResult bindingResult) {

if(bindingResult.hasErrors()){

return "movie/movie-edit";

}

movieRepository.save(movie);

return "redirect:/movie";

}

@GetMapping("/movie/show/{movie}")

public String movieShow(

Movie movie,

Model model) {

model.addAttribute("movie", movie);

return "movie/movie-show";

}

@GetMapping("/movie/del/{movie}")

public String movieDel(

Movie movie) {

movieRepository.delete(movie);

return "redirect:../";

}

@GetMapping("/movie/filter")

public String blogFilter(Model model) {

return "movie/movie-filter";

}

@PostMapping("/movie/filter/result")

public String blogResult(@RequestParam String title, Model model) {

List<Movie> result = movieRepository.findByNameContains(title);

model.addAttribute("result", result);

return "movie/movie-filter";

}

@PostMapping("/movie/filter\_strict/result")

public String blogStrictResult(@RequestParam String title, Model model) {

List<Movie> result = movieRepository.findByName(title);

model.addAttribute("result", result);

return "movie/movie-filter";

}

}

15. java\com\hludencov\java\_spring\controllers\Personal\_infoController.java

package com.hludencov.java\_spring.controllers;

import com.hludencov.java\_spring.models.Personal\_info;

import com.hludencov.java\_spring.repo.Personal\_infoRepository;

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

import org.springframework.security.access.prepost.PreAuthorize;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

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

import javax.validation.Valid;

import java.util.List;

@Controller

@RequestMapping("/personal\_info")

@PreAuthorize("hasAnyAuthority('ADMIN', 'TEACHER')")

public class Personal\_infoController {

@Autowired

private Personal\_infoRepository personal\_infoRepository;

@GetMapping

public String educationList(Personal\_info personal\_info, Model model){

model.addAttribute("personal\_info", personal\_infoRepository.findAll());

return "personal\_info/personal\_info-main";

}

@GetMapping("/add")

public String educationAdd(Personal\_info personal\_info) {

return "personal\_info/personal\_info-add";

}

@PostMapping("/add")

public String educationPostAdd(@ModelAttribute("personal\_info") @Valid Personal\_info personal\_info, BindingResult bindingResult) {

if (bindingResult.hasErrors()) {

return "personal\_info/personal\_info-add";

}

personal\_infoRepository.save(personal\_info);

return "redirect:/personal\_info";

}

@GetMapping("/edit/{personal\_info}")

public String educationEdit(

Personal\_info personal\_info,

Model model) {

model.addAttribute("personal\_info", personal\_info);

return "personal\_info/personal\_info-edit";

}

@PostMapping("/edit/{personal\_info}")

public String educationPostEdit(@ModelAttribute("personal\_info") @Valid Personal\_info personal\_info, BindingResult bindingResult) {

if (bindingResult.hasErrors()) {

return "personal\_info/personal\_info-edit";

}

personal\_infoRepository.save(personal\_info);

return "redirect:../";

}

@GetMapping("/show/{personal\_info}")

public String educationShow(

Personal\_info personal\_info,

Model model) {

model.addAttribute("personal\_info", personal\_info);

return "personal\_info/personal\_info-show";

}

@GetMapping("/del/{personal\_info}")

public String educationDel(

Personal\_info personal\_info) {

personal\_infoRepository.delete(personal\_info);

return "redirect:../";

}

@GetMapping("/filter")

public String educationFilter(Model model) {

return "personal\_info/personal\_info-filter";

}

@PostMapping("/filter/result")

public String educationResult(@RequestParam String title, Model model) {

List<Personal\_info> result = personal\_infoRepository.findByNameContains(title);

model.addAttribute("result", result);

return "personal\_info/personal\_info-filter";

}

@PostMapping("/filter\_strict/result")

public String educationStrictResult(@RequestParam String title, Model model) {

List<Personal\_info> result = personal\_infoRepository.findByName(title);

model.addAttribute("result", result);

return "personal\_info/personal\_info-filter";

}

}

16. java\com\hludencov\java\_spring\controllers\Preparation\_programController.java

package com.hludencov.java\_spring.controllers;

import com.hludencov.java\_spring.models.Preparation\_program;

import com.hludencov.java\_spring.repo.Preparation\_programRepository;

import com.hludencov.java\_spring.repo.SubjectRepository;

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

import org.springframework.security.access.prepost.PreAuthorize;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

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

import javax.validation.Valid;

import java.util.List;

@Controller

@RequestMapping("/preparation\_program")

@PreAuthorize("hasAuthority('ADMIN') or hasAuthority('TEACHER')")

public class Preparation\_programController {

@Autowired

private Preparation\_programRepository preparation\_programRepository;

@Autowired

private SubjectRepository subjectRepository;

@GetMapping

public String prepprogList(Preparation\_program preparation\_program, Model model){

model.addAttribute("preparation\_program", preparation\_programRepository.findAll());

return "preparation\_program/preparation\_program-main";

}

@GetMapping("/add")

public String prepprogAdd(Preparation\_program preparation\_program,Model model) {

model.addAttribute("subjectSet", subjectRepository.findAll());

return "preparation\_program/preparation\_program-add";

}

@PostMapping("/add")

public String prepprogPostAdd(@ModelAttribute("preparation\_program") @Valid Preparation\_program preparation\_program, BindingResult bindingResult, Model model) {

if (bindingResult.hasErrors()) {

model.addAttribute("subjectSet", subjectRepository.findAll());

return "preparation\_program/preparation\_program-add";

}

preparation\_programRepository.save(preparation\_program);

return "redirect:/preparation\_program";

}

@GetMapping("/edit/{preparation\_program}")

public String prepprogEdit(

Preparation\_program preparation\_program,

Model model) {

model.addAttribute("subjectSet", subjectRepository.findAll());

model.addAttribute("preparation\_program", preparation\_program);

return "preparation\_program/preparation\_program-edit";

}

@PostMapping("/edit/{preparation\_program}")

public String prepprogPostEdit(@ModelAttribute("preparation\_program") @Valid Preparation\_program preparation\_program, BindingResult bindingResult, Model model) {

if (bindingResult.hasErrors()) {

model.addAttribute("subjectSet", subjectRepository.findAll());

return "preparation\_program/preparation\_program-edit";

}

preparation\_programRepository.save(preparation\_program);

return "redirect:../";

}

@GetMapping("/show/{preparation\_program}")

public String prepprogShow(

Preparation\_program preparation\_program,

Model model) {

model.addAttribute("preparation\_program", preparation\_program);

return "preparation\_program/preparation\_program-show";

}

@GetMapping("/del/{preparation\_program}")

public String prepprogDel(

Preparation\_program preparation\_program) {

preparation\_programRepository.delete(preparation\_program);

return "redirect:../";

}

@GetMapping("/filter")

public String prepprogFilter(Model model) {

return "preparation\_program/preparation\_program-filter";

}

@PostMapping("/filter/result")

public String prepprogResult(@RequestParam String title, Model model) {

List<Preparation\_program> result = preparation\_programRepository.findByNameContains(title);

model.addAttribute("result", result);

return "preparation\_program/preparation\_program-filter";

}

@PostMapping("/filter\_strict/result")

public String prepprogStrictResult(@RequestParam String title, Model model) {

List<Preparation\_program> result = preparation\_programRepository.findByName(title);

model.addAttribute("result", result);

return "preparation\_program/preparation\_program-filter";

}

}

17. java\com\hludencov\java\_spring\controllers\SubjectController.java

package com.hludencov.java\_spring.controllers;

import com.hludencov.java\_spring.models.Subject;

import com.hludencov.java\_spring.repo.SubjectRepository;

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

import org.springframework.security.access.prepost.PreAuthorize;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

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

import javax.validation.Valid;

import java.util.List;

@Controller

@RequestMapping("/subject")

@PreAuthorize("hasAnyAuthority('ADMIN')")

public class SubjectController {

@Autowired

private SubjectRepository subjectRepository;

@GetMapping

public String subjectList(Model model){

model.addAttribute("subject", subjectRepository.findAll());

return "subject/subject-main";

}

@GetMapping("/add")

public String subjectAdd(Subject subject) {

return "subject/subject-add";

}

@PostMapping("/add")

public String subjectPostAdd(@ModelAttribute("subject") @Valid Subject subject, BindingResult bindingResult) {

if (bindingResult.hasErrors()) {

return "subject/subject-add";

}

subjectRepository.save(subject);

return "redirect:/subject";

}

@GetMapping("/edit/{subject}")

public String subjectEdit(

Subject subject,

Model model) {

model.addAttribute("subject", subject);

return "subject/subject-edit";

}

@PostMapping("/edit/{subject}")

public String subjectPostEdit(@ModelAttribute("subject") @Valid Subject subject, BindingResult bindingResult) {

if (bindingResult.hasErrors()) {

return "subject/subject-edit";

}

subjectRepository.save(subject);

return "redirect:../";

}

@GetMapping("/show/{subject}")

public String subjectShow(

Subject subject,

Model model) {

model.addAttribute("subject", subject);

return "subject/subject-show";

}

@GetMapping("/del/{subject}")

public String subjectDel(

Subject subject) {

subjectRepository.delete(subject);

return "redirect:../";

}

@GetMapping("/filter")

public String subjectFilter(Model model) {

return "subject/subject-filter";

}

@PostMapping("/filter/result")

public String subjectResult(@RequestParam String title, Model model) {

List<Subject> result = subjectRepository.findByNameContains(title);

model.addAttribute("result", result);

return "subject/subject-filter";

}

@PostMapping("/filter\_strict/result")

public String subjectStrictResult(@RequestParam String title, Model model) {

List<Subject> result = subjectRepository.findByName(title);

model.addAttribute("result", result);

return "subject/subject-filter";

}

}

18. java\com\hludencov\java\_spring\controllers\SummaryController.java

package com.hludencov.java\_spring.controllers;

import com.hludencov.java\_spring.models.Summary;

import com.hludencov.java\_spring.repo.CandidateRepository;

import com.hludencov.java\_spring.repo.Education\_institutionRepository;

import com.hludencov.java\_spring.repo.SubjectRepository;

import com.hludencov.java\_spring.repo.SummaryRepository;

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

import org.springframework.security.access.prepost.PreAuthorize;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

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

import javax.validation.Valid;

import java.util.List;

@Controller

@RequestMapping("/summary")

@PreAuthorize("hasAuthority('TEACHER') or hasAuthority('USER')")

public class SummaryController {

@Autowired

private SummaryRepository summaryRepository;

@Autowired

private Education\_institutionRepository education\_institutionRepository;

@Autowired

private SubjectRepository subjectRepository;

@Autowired

private CandidateRepository candidateRepository;

@GetMapping

public String summaryList(Model model) {

model.addAttribute("summary", summaryRepository.findAll());

model.addAttribute("subjects", subjectRepository.findAll());

model.addAttribute("candidate", candidateRepository.findAll());

return "summary/summary-main";

}

@GetMapping("/add")

public String summaryAdd(Summary summary, Model model) {

model.addAttribute("education\_institution", education\_institutionRepository.findAll());

model.addAttribute("subjects", subjectRepository.findAll());

model.addAttribute("candidate", candidateRepository.findAll());

return "summary/summary-add";

}

@PostMapping("/add")

public String summaryPostAdd(@ModelAttribute("summary") @Valid Summary summary, BindingResult bindingResult, Model model) {

if (bindingResult.hasErrors()) {

model.addAttribute("education\_institution", education\_institutionRepository.findAll());

model.addAttribute("subjects", subjectRepository.findAll());

model.addAttribute("candidate", candidateRepository.findAll());

return "summary/summary-add";

}

summaryRepository.save(summary);

return "redirect:/summary";

}

@GetMapping("/edit/{summary}")

public String summaryEdit(

Summary summary,

Model model) {

model.addAttribute("candidate", candidateRepository.findAll());

model.addAttribute("education\_institution", education\_institutionRepository.findAll());

model.addAttribute("subjects", subjectRepository.findAll());

model.addAttribute("candidate", candidateRepository.findAll());

model.addAttribute("summary", summary);

return "summary/summary-edit";

}

@PostMapping("/edit/{summary}")

public String summaryPostEdit(@ModelAttribute("summary") @Valid Summary summary, BindingResult bindingResult, Model model) {

if (bindingResult.hasErrors()) {

model.addAttribute("education\_institution", education\_institutionRepository.findAll());

model.addAttribute("subjects", subjectRepository.findAll());

return "summary/summary-edit";

}

summaryRepository.save(summary);

return "redirect:../";

}

@GetMapping("/show/{summary}")

public String summaryShow(

Summary summary,

Model model) {

model.addAttribute("summary", summary);

return "summary/summary-show";

}

@GetMapping("/del/{summary}")

public String summaryDel(

Summary summary) {

summaryRepository.delete(summary);

return "redirect:../";

}

@GetMapping("/filter")

public String summaryFilter(Model model) {

return "summary/summary-filter";

}

@PostMapping("/filter/result")

public String summaryResult(@RequestParam String title, Model model) {

List<Summary> result = summaryRepository.findByMarkContains(title);

model.addAttribute("result", result);

return "summary/summary-filter";

}

}

19. java\com\hludencov\java\_spring\controllers\TeacherInfoController.java

package com.hludencov.java\_spring.controllers;

import com.hludencov.java\_spring.models.Teacher\_info;

import com.hludencov.java\_spring.repo.SubjectRepository;

import com.hludencov.java\_spring.repo.TeachersRepository;

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

import org.springframework.security.access.prepost.PreAuthorize;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

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

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

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

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

import javax.validation.Valid;

@Controller

@RequestMapping("/teacher")

@PreAuthorize("hasAnyAuthority('TEACHER')")

public class TeacherInfoController {

@Autowired

private TeachersRepository teachersRepository;

@Autowired

private SubjectRepository subjectRepository;

@GetMapping

public String teacherList(Model model) {

model.addAttribute("teacher", teachersRepository.findAll());

return "teacher/teacher-main";

}

@GetMapping("/add")

public String teacherAdd(Teacher\_info teacher, Model model) {

model.addAttribute("teacher", teachersRepository.findAll());

model.addAttribute("subjects", subjectRepository.findAll());

return "teacher/teacher-add";

}

@PostMapping("/add")

public String teacherPostAdd(@ModelAttribute("teacher") @Valid Teacher\_info teacher, BindingResult bindingResult, Model model) {

if (bindingResult.hasErrors()) {

model.addAttribute("teacher", teachersRepository.findAll());

model.addAttribute("subjects", subjectRepository.findAll());

return "teacher/teacher-add";

}

teachersRepository.save(teacher);

return "redirect:/teacher";

}

@GetMapping("/edit/{teacher}")

public String teacherEdit(

Teacher\_info teacher,

Model model) {

model.addAttribute("teacher", teachersRepository.findAll());

model.addAttribute("subjects", subjectRepository.findAll());

model.addAttribute("teacher", teacher);

return "teacher/teacher-edit";

}

@PostMapping("/edit/{teacher}")

public String teacherPostEdit(@ModelAttribute("teacher") @Valid Teacher\_info teacher, BindingResult bindingResult, Model model) {

if (bindingResult.hasErrors()) {

model.addAttribute("teacher", teachersRepository.findAll());

model.addAttribute("subjects", subjectRepository.findAll());

return "teacher/teacher-edit";

}

teachersRepository.save(teacher);

return "redirect:../";

}

@GetMapping("/del/{subject}")

public String teachertDel(

Teacher\_info teacher) {

teachersRepository.delete(teacher);

return "redirect:../";

}

}

20. java\com\hludencov\java\_spring\controllers\UsersController.java

package com.hludencov.java\_spring.controllers;

import com.hludencov.java\_spring.models.Role;

import com.hludencov.java\_spring.models.User;

import com.hludencov.java\_spring.repo.UsersRepository;

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

import org.springframework.security.access.prepost.PreAuthorize;

import org.springframework.security.crypto.password.PasswordEncoder;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

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

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

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

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

import javax.validation.Valid;

import java.util.List;

@Controller

@PreAuthorize("hasAnyAuthority('ADMIN')")

public class UsersController {

@Autowired

UsersRepository usersRepository;

@Autowired

private PasswordEncoder passwordEncoder;

@GetMapping("/user")

public String userMain(Model model) {

Iterable<User> users = usersRepository.findAll();

model.addAttribute("users", users);

return "user/user-main";

}

@GetMapping("/user/add")

public String userAdd(User user, Model model) {

model.addAttribute("roles", Role.values());

return "user/user-add";

}

@PostMapping("/user/add")

public String userPostAdd(@ModelAttribute("user") @Valid User user, BindingResult bindingResult) {

if (bindingResult.hasErrors()) {

return "user/user-add";

}

user.setActive(true);

user.setPassword(passwordEncoder.encode(user.getPassword()));

usersRepository.save(user);

return "redirect:/user";

}

@GetMapping("/user/edit/{user}")

public String userEdit(

User user,

Model model) {

model.addAttribute("user", user);

model.addAttribute("roles", Role.values());

return "user/user-edit";

}

@PostMapping("/user/edit/{user}")

public String userPostEdit(@ModelAttribute("user") @Valid User user, BindingResult bindingResult) {

if (bindingResult.hasErrors()) {

return "user/user-edit";

}

user.setActive(true);

user.setPassword(passwordEncoder.encode(user.getPassword()));

usersRepository.save(user);

return "redirect:../";

}

@GetMapping("/user/show/{user}")

public String userShow(

User user,

Model model) {

model.addAttribute("user", user);

return "user/user-show";

}

@GetMapping("/user/del/{user}")

public String userDel(

User user) {

usersRepository.delete(user);

return "redirect:../";

}

@GetMapping("/user/filter")

public String userFilter(Model model) {

return "user/user-filter";

}

@PostMapping("/user/filter/result")

public String userResult(@RequestParam String title, Model model) {

List<User> result = usersRepository.findByLoginContains(title);

model.addAttribute("result", result);

return "user/user-filter";

}

@PostMapping("/user/filter\_strict/result")

public String userStrictResult(@RequestParam String title, Model model) {

List<User> result = usersRepository.findByLogin(title);

model.addAttribute("result", result);

return "user/user-filter";

}

}

21. java\com\hludencov\java\_spring\models\Candidate\_info.java

package com.hludencov.java\_spring.models;

import javax.persistence.\*;

import javax.validation.constraints.\*;

import java.sql.Date;

import java.util.Set;

@Entity

@Table(name = "candidate\_info")

public class Candidate\_info {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

public Long id;

@PastOrPresent

public Date submissionDate;

@ManyToOne

@JoinColumn(name = "target\_department\_id")

public Department target\_department;

@OneToOne(cascade = CascadeType.ALL)

@JoinColumn(name = "user\_id")

public User user;

@OneToMany(mappedBy = "candidate\_info", fetch = FetchType.EAGER)

public Set<Summary> summaries;

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_BOILERPLATE LINE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public User getUser() {

return user;

}

public void setUser(User user) {

this.user = user;

}

public Department getTarget\_department() {

return target\_department;

}

public void setTarget\_department(Department target\_department) {

this.target\_department = target\_department;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public Date getSubmissionDate() {

return submissionDate;

}

public void setSubmissionDate(Date submission\_date) {

this.submissionDate = submission\_date;

}

public Set<Summary> getSummaries() {

return summaries;

}

public void setSummaries(Set<Summary> summaries) {

this.summaries = summaries;

}

}

22. java\com\hludencov\java\_spring\models\Department.java

package com.hludencov.java\_spring.models;

import javax.persistence.\*;

import javax.validation.constraints.\*;

import java.util.List;

import java.util.Set;

@Entity

@Table(name = "Department")

public class Department {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@NotBlank

private String name;

@OneToMany(mappedBy = "target\_department", fetch = FetchType.EAGER)

private Set<Candidate\_info> candidate\_infos;

@OneToMany(mappedBy = "department", fetch = FetchType.EAGER)

private Set<User> department\_users;

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_BOILERPLATE LINE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public Set<Candidate\_info> getCandidate\_infos() {

return candidate\_infos;

}

public void setCandidate\_infos(Set<Candidate\_info> candidate\_infos) {

this.candidate\_infos = candidate\_infos;

}

public Set<User> getDepartment\_users() {

return department\_users;

}

public void setDepartment\_users(Set<User> department\_users) {

this.department\_users = department\_users;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

23. java\com\hludencov\java\_spring\models\Document.java

package com.hludencov.java\_spring.models;

import javax.persistence.\*;

import javax.validation.constraints.\*;

import java.sql.Date;

import java.util.Set;

@Entity

public class Document {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

public Long id;

@ManyToOne(optional = true)

@JoinColumn(name = "user\_id")

public User user;

@NotBlank

public String fileName;

@PastOrPresent

public Date date;

@FutureOrPresent

public Date archive\_date;

@OneToMany(mappedBy = "document", fetch = FetchType.EAGER)

private Set<Summary> summaries;

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_BOILERPLATE LINE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public Set<Summary> getSummaries() {

return summaries;

}

public void setSummaries(Set<Summary> summaries) {

this.summaries = summaries;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public User getUser() {

return user;

}

public void setUser(User user) {

this.user = user;

}

public String getFileName() {

return fileName;

}

public void setFileName(String fileName) {

this.fileName = fileName;

}

public Date getDate() {

return date;

}

public void setDate(Date date) {

this.date = date;

}

public Date getArchive\_date() {

return archive\_date;

}

public void setArchive\_date(Date archive\_date) {

this.archive\_date = archive\_date;

}

}

24. java\com\hludencov\java\_spring\models\Education\_institution.java

package com.hludencov.java\_spring.models;

import javax.persistence.\*;

import javax.validation.constraints.NotBlank;

import java.util.Set;

@Entity

@Table(name = "education\_institution")

public class Education\_institution {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@NotBlank

public String name, address;

@OneToMany(mappedBy = "education\_institution", fetch = FetchType.EAGER)

private Set<Summary> summaries;

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_BOILERPLATE LINE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public Set<Summary> getSummaries() {

return summaries;

}

public void setSummaries(Set<Summary> summaries) {

this.summaries = summaries;

}

}

25. java\com\hludencov\java\_spring\models\Genre.java

package com.hludencov.java\_spring.models;

import javax.persistence.\*;

import javax.validation.constraints.NotBlank;

import javax.validation.constraints.Size;

import java.util.List;

@Entity

public class Genre {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

public Long id;

@NotBlank

@Size(min = 1, max = 70)

public String name;

@ManyToMany

@JoinTable (name="genres\_movie",

joinColumns=@JoinColumn (name="movie\_id"),

inverseJoinColumns=@JoinColumn(name="genre\_id"))

public List<Movie> movie;

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public List<Movie> getMovie() {

return movie;

}

public void setMovie(List<Movie> movie) {

this.movie = movie;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

26. java\com\hludencov\java\_spring\models\Group.java

package com.hludencov.java\_spring.models;

import javax.persistence.\*;

import java.util.Set;

@Entity

@Table(name = "groups")

public class Group {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@ManyToMany

@JoinTable(name = "user\_to\_group",

joinColumns = @JoinColumn(name = "groups\_id"),

inverseJoinColumns = @JoinColumn(name = "user\_id"))

public Set<User> users;

@ManyToOne

@JoinColumn(name = "user\_id")

public User teacher\_organizer;

@ManyToOne

@JoinColumn(name = "preparation\_program\_id")

public Preparation\_program preparation\_program;

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_BOILERPLATE LINE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public Set<User> getUsers() {

return users;

}

public void setUsers(Set<User> users) {

this.users = users;

}

public Preparation\_program getPreparation\_program() {

return preparation\_program;

}

public void setPreparation\_program(Preparation\_program preparation\_program) {

this.preparation\_program = preparation\_program;

}

public User getTeacher\_organizer() {

return teacher\_organizer;

}

public void setTeacher\_organizer(User teacher\_organizer) {

this.teacher\_organizer = teacher\_organizer;

}

}

27. java\com\hludencov\java\_spring\models\Movie.java

package com.hludencov.java\_spring.models;

import javax.persistence.\*;

import javax.validation.constraints.\*;

import java.sql.Date;

import java.util.List;

@Entity

@Table (name = "movies")

public class Movie {

public Movie(String name,Date release\_date,double rating,int series\_amount, int comments\_amount){

this.name=name;

this.release\_date=release\_date;

this.rating=rating;

this.series\_amount=series\_amount;

this.comments\_amount=comments\_amount;

}

public Movie(){}

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@NotNull

@NotBlank

private String name;

@NotNull(message = "ыыыыедите корректную дату")

private Date release\_date;

@NotNull

@PositiveOrZero

private double rating;

@NotNull

@PositiveOrZero

private int series\_amount, comments\_amount;

@OneToMany(mappedBy = "movie", fetch = FetchType.EAGER)

private List<Post> post;

@ManyToMany()

@JoinTable(name = "genres\_movie",

joinColumns = @JoinColumn (name = "genre\_id"),

inverseJoinColumns = @JoinColumn (name = "movie\_id")

)

public List<Genre> genres;

public int getSeries\_amount() {

return series\_amount;

}

public void setSeries\_amount(int series\_amount) {

this.series\_amount = series\_amount;

}

public int getComments\_amount() {

return comments\_amount;

}

public void setComments\_amount(int comments\_amount) {

this.comments\_amount = comments\_amount;

}

public double getRating() {

return rating;

}

public void setRating(double rating) {

this.rating = rating;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Date getRelease\_date() {

return release\_date;

}

public void setRelease\_date(Date release\_date) {

this.release\_date = release\_date;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

}

28. java\com\hludencov\java\_spring\models\Personal\_info.java

package com.hludencov.java\_spring.models;

import org.springframework.lang.Nullable;

import javax.persistence.\*;

import javax.validation.constraints.NotBlank;

import javax.validation.constraints.NotNull;

import javax.validation.constraints.PastOrPresent;

import java.sql.Date;

import java.util.Set;

@Entity

@Table(name = "personal\_info")

public class Personal\_info {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@NotBlank

private String name, surname;

@Nullable

private String sec\_name;

@PastOrPresent

private Date birthdate;

@NotNull

private int pass\_id;

@NotNull

private boolean is\_male;

@OneToOne(cascade = CascadeType.ALL)

@JoinColumn(name = "user\_id")

private User user;

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_BOILERPLATE LINE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public User getUser() {

return user;

}

public void setUser(User user) {

this.user = user;

}

public int getPass\_id() {

return pass\_id;

}

public void setPass\_id(int pass\_id) {

this.pass\_id = pass\_id;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getSurname() {

return surname;

}

public void setSurname(String surname) {

this.surname = surname;

}

@Nullable

public String getSec\_name() {

return sec\_name;

}

public void setSec\_name(@Nullable String sec\_name) {

this.sec\_name = sec\_name;

}

public Date getBirthdate() {

return birthdate;

}

public void setBirthdate(Date birthdate) {

this.birthdate = birthdate;

}

public boolean isIs\_male() {

return is\_male;

}

public void setIs\_male(boolean is\_male) {

this.is\_male = is\_male;

}

}

29. java\com\hludencov\java\_spring\models\Post.java

package com.hludencov.java\_spring.models;

import javax.persistence.\*;

import java.util.List;

@Entity

@Table(name = "post")

public class Post {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String title, anons, full\_text;

private int views;

@OneToOne(cascade = CascadeType.ALL)

@JoinColumn(name = "user\_id")

public User user;

@ManyToOne(optional = true, cascade = CascadeType.ALL)

public Movie movie;

public User getUser() {

return user;

}

public void setUser(User user) {

this.user = user;

}

public Movie getMovie() {

return movie;

}

public void setMovie(Movie movie) {

this.movie = movie;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getAnons() {

return anons;

}

public void setAnons(String anons) {

this.anons = anons;

}

public String getFull\_text() {

return full\_text;

}

public void setFull\_text(String full\_text) {

this.full\_text = full\_text;

}

public int getViews() {

return views;

}

public void setViews(int views) {

this.views = views;

}

}

30. java\com\hludencov\java\_spring\models\Preparation\_program.java

package com.hludencov.java\_spring.models;

import javax.persistence.\*;

import javax.validation.constraints.\*;

import java.util.Set;

@Entity

@Table(name = "preparation\_program")

public class Preparation\_program {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@NotBlank

private String name;

@ManyToMany

@JoinTable(name = "prep\_to\_subject",

joinColumns = @JoinColumn(name = "preparation\_program\_id"),

inverseJoinColumns = @JoinColumn(name = "subject\_id"))

public Set<Subject> subjectSet;

@OneToMany(mappedBy = "preparation\_program", fetch = FetchType.EAGER)

private Set<Group> groups;

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_BOILERPLATE LINE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public Set<Group> getGroups() {

return groups;

}

public void setGroups(Set<Group> groups) {

this.groups = groups;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Set<Subject> getSubjectSet() {

return subjectSet;

}

public void setSubjectSet(Set<Subject> subjectSet) {

this.subjectSet = subjectSet;

}

}

31. java\com\hludencov\java\_spring\models\Role.java

package com.hludencov.java\_spring.models;

import org.springframework.security.core.GrantedAuthority;

import java.util.List;

public enum Role implements GrantedAuthority {

USER, ADMIN, CANDIDATE, TEACHER;

@Override

public String getAuthority() {

return name();

}

}

32. java\com\hludencov\java\_spring\models\Subject.java

package com.hludencov.java\_spring.models;

import javax.persistence.\*;

import javax.validation.constraints.NotBlank;

import java.util.Set;

@Entity

@Table(name = "subject")

public class Subject {

public Subject(Long id, String subject\_code, String name, Set<Teacher\_info> teacher\_infoList) {

this.id = id;

this.subject\_code = subject\_code;

this.name = name;

this.teacher\_infoList = teacher\_infoList;

}

public Subject() {

}

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@NotBlank

public String subject\_code;

@NotBlank

public String name;

@ManyToMany

@JoinTable(name = "teacher\_to\_subject",

joinColumns = @JoinColumn(name = "teacher\_info\_id"),

inverseJoinColumns = @JoinColumn(name = "subject\_id"))

public Set<Teacher\_info> teacher\_infoList;

@ManyToMany

@JoinTable(name = "prep\_to\_subject",

joinColumns = @JoinColumn(name = "subject\_id"),

inverseJoinColumns = @JoinColumn(name = "preparation\_program\_id"))

public Set<Preparation\_program> preparation\_programs;

@OneToMany(mappedBy = "subject", fetch = FetchType.EAGER)

private Set<Summary> summaries;

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_BOILERPLATE LINE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public Set<Preparation\_program> getPreparation\_programs() {

return preparation\_programs;

}

public void setPreparation\_programs(Set<Preparation\_program> preparation\_programs) {

this.preparation\_programs = preparation\_programs;

}

public Set<Summary> getSummaries() {

return summaries;

}

public void setSummaries(Set<Summary> summaries) {

this.summaries = summaries;

}

public Set<Teacher\_info> getTeacher\_infoList() {

return teacher\_infoList;

}

public void setTeacher\_infoList(Set<Teacher\_info> teacher\_infoList) {

this.teacher\_infoList = teacher\_infoList;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getSubject\_code() {

return subject\_code;

}

public void setSubject\_code(String subject\_code) {

this.subject\_code = subject\_code;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

33. java\com\hludencov\java\_spring\models\Summary.java

package com.hludencov.java\_spring.models;

import javax.persistence.Entity;

import javax.persistence.\*;

import javax.validation.constraints.\*;

@Entity

public class Summary {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@NotNull

@Min(1)

@Max(5)

private int mark;

@ManyToOne

@JoinColumn(name = "document\_id")

private Document document;

@ManyToOne

@JoinColumn(name = "subject\_id")

private Subject subject;

@ManyToOne

@JoinColumn(name = "education\_institution\_id")

private Education\_institution education\_institution;

@ManyToOne

@JoinColumn(name = "candidate\_info\_id")

private Candidate\_info candidate\_info;

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_BOILERPLATE LINE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public Education\_institution getEducation\_institution() {

return education\_institution;

}

public void setEducation\_institution(Education\_institution education\_institution) {

this.education\_institution = education\_institution;

}

public Subject getSubject() {

return subject;

}

public void setSubject(Subject subject) {

this.subject = subject;

}

public Document getDocument() {

return document;

}

public void setDocument(Document document) {

this.document = document;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public int getMark() {

return mark;

}

public void setMark(int mark) {

this.mark = mark;

}

public Candidate\_info getCandidate\_info() {

return candidate\_info;

}

public void setCandidate\_info(Candidate\_info candidate\_info) {

this.candidate\_info = candidate\_info;

}

}

34. java\com\hludencov\java\_spring\models\Teacher\_info.java

package com.hludencov.java\_spring.models;

import javax.persistence.\*;

import javax.validation.constraints.\*;

import java.sql.Date;

import java.util.Set;

@Entity

@Table(name = "teacher\_info")

public class Teacher\_info {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@PastOrPresent

public Date teachSince;

@ManyToMany

@JoinTable(name = "teacher\_to\_subject",

joinColumns = @JoinColumn(name = "subject\_id"),

inverseJoinColumns = @JoinColumn(name = "teacher\_info\_id"))

public Set<Subject> subjectList;

@OneToMany(mappedBy = "teacher\_info", fetch = FetchType.EAGER)

public Set<User> user\_teacher\_info;

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_BOILERPLATE LINE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public Set<User> getUser\_teacher\_info() {

return user\_teacher\_info;

}

public void setUser\_teacher\_info(Set<User> user\_teacher\_info) {

this.user\_teacher\_info = user\_teacher\_info;

}

public Set<Subject> getSubjectList() {

return subjectList;

}

public void setSubjectList(Set<Subject> subjectList) {

this.subjectList = subjectList;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public Date getTeachSince() {

return teachSince;

}

public void setTeachSince(Date teachSince) {

this.teachSince = teachSince;

}

}

35. java\com\hludencov\java\_spring\models\User.java

package com.hludencov.java\_spring.models;

import org.springframework.security.core.GrantedAuthority;

import org.springframework.security.core.userdetails.UserDetails;

import javax.persistence.\*;

import javax.validation.constraints.\*;

import java.sql.Date;

import java.util.Collection;

import java.util.Set;

@Entity

@Table(name = "user")

public class User implements UserDetails {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@NotBlank

private String login;

@NotBlank

private String password;

@OneToMany(mappedBy = "user", fetch = FetchType.EAGER)

private Set<Document> documents;

private boolean active;

@ElementCollection(targetClass = Role.class, fetch = FetchType.EAGER)

@CollectionTable(name = "user\_role", joinColumns = @JoinColumn(name = "user\_id"))

@Enumerated(EnumType.STRING)

private Set<Role> roles;

@ManyToMany

@JoinTable(name = "user\_to\_group",

joinColumns = @JoinColumn(name = "user\_id"),

inverseJoinColumns = @JoinColumn(name = "groups\_id"))

private Set<Group> groups;

@ManyToOne

@JoinColumn(name = "department\_id")

private Department department;

@OneToOne(mappedBy = "user")

private Personal\_info personal\_info;

@OneToOne(mappedBy = "user")

private Candidate\_info candidate\_info;

@ManyToOne

@JoinColumn(name = "teacher\_info\_id")

private Teacher\_info teacher\_info;

@OneToMany(mappedBy = "teacher\_organizer", fetch = FetchType.EAGER)

private Set<Group> teacher\_org\_groups;

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_BOILERPLATE LINE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public Set<Group> getTeacher\_org\_groups() {

return teacher\_org\_groups;

}

public void setTeacher\_org\_groups(Set<Group> teacher\_org\_groups) {

this.teacher\_org\_groups = teacher\_org\_groups;

}

public Teacher\_info getTeacher\_info() {

return teacher\_info;

}

public void setTeacher\_info(Teacher\_info teacher\_info) {

this.teacher\_info = teacher\_info;

}

public Candidate\_info getCandidate\_info() {

return candidate\_info;

}

public void setCandidate\_info(Candidate\_info candidate\_info) {

this.candidate\_info = candidate\_info;

}

public Personal\_info getPersonal\_info() {

return personal\_info;

}

public void setPersonal\_info(Personal\_info personal\_info) {

this.personal\_info = personal\_info;

}

public Department getDepartment() {

return department;

}

public void setDepartment(Department department) {

this.department = department;

}

public Set<Group> getGroups() {

return groups;

}

public void setGroups(Set<Group> groups) {

this.groups = groups;

}

public Set<Document> getDocuments() {

return documents;

}

public void setDocuments(Set<Document> documents) {

this.documents = documents;

}

public void setId(Long id) {

this.id = id;

}

public void setLogin(String nickname) {

this.login = nickname;

}

public Long getId() {

return id;

}

public boolean isActive() {

return active;

}

public void setActive(boolean active) {

this.active = active;

}

public Set<Role> getRoles() {

return roles;

}

public void setRoles(Set<Role> roles) {

this.roles = roles;

}

@Override

public Collection<? extends GrantedAuthority> getAuthorities() {

return getRoles();

}

public String getPassword() {

return password;

}

@Override

public String getUsername() {

return getLogin();

}

public String getLogin() {

return this.login;

}

@Override

public boolean isAccountNonExpired() {

return true;

}

@Override

public boolean isAccountNonLocked() {

return true;

}

@Override

public boolean isCredentialsNonExpired() {

return true;

}

@Override

public boolean isEnabled() {

return isActive();

}

public void setPassword(String password) {

this.password = password;

}

}

36. java\com\hludencov\java\_spring\repo\CandidateRepository.java

package com.hludencov.java\_spring.repo;

import com.hludencov.java\_spring.models.Candidate\_info;

import org.springframework.data.repository.CrudRepository;

import java.util.List;

public interface CandidateRepository extends CrudRepository<Candidate\_info, Long> {

List<Candidate\_info> findBySubmissionDateContains(String name);

}

37. java\com\hludencov\java\_spring\repo\DepartmentRepository.java

package com.hludencov.java\_spring.repo;

import com.hludencov.java\_spring.models.Department;

import org.springframework.data.repository.CrudRepository;

import java.util.List;

public interface DepartmentRepository extends CrudRepository<Department, Long> {

List<Department> findByNameContains(String name);

List<Department> findByName(String name);

}

38. java\com\hludencov\java\_spring\repo\DocumentRepository.java

package com.hludencov.java\_spring.repo;

import com.hludencov.java\_spring.models.Document;

import org.springframework.data.repository.CrudRepository;

import java.util.List;

public interface DocumentRepository extends CrudRepository<Document, Long> {

List<Document> findByFileNameContains(String name);

List<Document> findByFileName(String name);

}

39. java\com\hludencov\java\_spring\repo\Education\_institutionRepository.java

package com.hludencov.java\_spring.repo;

import com.hludencov.java\_spring.models.Education\_institution;

import org.springframework.data.repository.CrudRepository;

import java.util.List;

public interface Education\_institutionRepository extends CrudRepository<Education\_institution, Long> {

List<Education\_institution> findByNameContains(String name);

List<Education\_institution> findByName(String title);

}

40. java\com\hludencov\java\_spring\repo\GenreRepository.java

package com.hludencov.java\_spring.repo;

import com.hludencov.java\_spring.models.Genre;

import org.springframework.data.repository.CrudRepository;

public interface GenreRepository extends CrudRepository<Genre, Long> {

}

41. java\com\hludencov\java\_spring\repo\GroupRepository.java

package com.hludencov.java\_spring.repo;

import com.hludencov.java\_spring.models.Group;

import org.springframework.data.repository.CrudRepository;

public interface GroupRepository extends CrudRepository<Group, Long> {

}

42. java\com\hludencov\java\_spring\repo\MovieRepository.java

package com.hludencov.java\_spring.repo;

import com.hludencov.java\_spring.models.Movie;

import org.springframework.data.repository.CrudRepository;

import java.util.List;

public interface MovieRepository extends CrudRepository<Movie, Long> {

List<Movie> findByNameContains(String title);

List<Movie> findByName(String title);

}

43. java\com\hludencov\java\_spring\repo\Personal\_infoRepository.java

package com.hludencov.java\_spring.repo;

import com.hludencov.java\_spring.models.Personal\_info;

import org.springframework.data.repository.CrudRepository;

import java.util.List;

public interface Personal\_infoRepository extends CrudRepository<Personal\_info, Long> {

List<Personal\_info> findByNameContains(String name);

List<Personal\_info> findByName(String title);

}

44. java\com\hludencov\java\_spring\repo\PostRepository.java

package com.hludencov.java\_spring.repo;

import com.hludencov.java\_spring.models.Post;

import org.springframework.data.repository.CrudRepository;

import java.util.List;

public interface PostRepository extends CrudRepository<Post, Long> {

List<Post> findByTitleContains(String title);

List<Post> findByTitle(String title);

}

45. java\com\hludencov\java\_spring\repo\Preparation\_programRepository.java

package com.hludencov.java\_spring.repo;

import com.hludencov.java\_spring.models.Preparation\_program;

import org.springframework.data.repository.CrudRepository;

import java.util.List;

public interface Preparation\_programRepository extends CrudRepository<Preparation\_program, Long> {

List<Preparation\_program> findByNameContains(String name);

List<Preparation\_program> findByName(String title);

}

46. java\com\hludencov\java\_spring\repo\SubjectRepository.java

package com.hludencov.java\_spring.repo;

import com.hludencov.java\_spring.models.Subject;

import org.springframework.data.repository.CrudRepository;

import java.util.List;

public interface SubjectRepository extends CrudRepository<Subject, Long> {

List<Subject> findByNameContains(String name);

List<Subject> findByName(String title);

}

47. java\com\hludencov\java\_spring\repo\SummaryRepository.java

package com.hludencov.java\_spring.repo;

import com.hludencov.java\_spring.models.Summary;

import org.springframework.data.repository.CrudRepository;

import java.util.List;

public interface SummaryRepository extends CrudRepository<Summary, Long> {

List<Summary> findByMarkContains(String title);

}

48. java\com\hludencov\java\_spring\repo\TeachersRepository.java

package com.hludencov.java\_spring.repo;

import com.hludencov.java\_spring.models.Teacher\_info;

import org.springframework.data.repository.CrudRepository;

import java.util.List;

public interface TeachersRepository extends CrudRepository<Teacher\_info, Long> {

List<Teacher\_info> findByTeachSinceContains(String title);

}

49. java\com\hludencov\java\_spring\repo\UserRepository.java

package com.hludencov.java\_spring.repo;

import com.hludencov.java\_spring.models.User;

import org.springframework.data.repository.CrudRepository;

public interface UserRepository extends CrudRepository<User, Long> {

User findByLogin(String title);

}

50. java\com\hludencov\java\_spring\repo\UsersRepository.java

package com.hludencov.java\_spring.repo;

import com.hludencov.java\_spring.models.User;

import org.springframework.data.repository.CrudRepository;

import java.util.List;

public interface UsersRepository extends CrudRepository<User, Long> {

List<User> findByLoginContains(String title);

List<User> findByLogin(String title);

}

51. resources\application.properties

spring.jpa.hibernate.ddl-auto=update

spring.datasource.url=jdbc:mysql://${MYSQL\_HOST:localhost}:3306/spring\_web\_blog

spring.datasource.username=root

spring.datasource.password=

spring.main.allow-circular-references=true

server.port=8080

52. resources\templates\blog\blog-add.html

<!DOCTYPE HTML>

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Добавление новой статьи</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

</head>

<body>

<div th:insert="header">...</div>

<form th:action="${'/blog/add'}" th:object="${post}" method="post">

<input type="text" name="title" placeholder="Введите название статьи">

<input type="text" name="anons" placeholder="Введите анос статьи">

<br>

<select th:field="\*{user}">

<option th:each="element : ${users}" th:text="${element.username}" th:value="${element.id}"></option>

</select>

<select th:field="\*{movie}">

<option th:each="element : ${movie}" th:text="${element.name}" th:value="${element.id}"></option>

</select>

<br>

<textarea name="full\_text" placeholder="Введите текст статьи"></textarea>

<br>

<button type="submit">Добавить статью</button>

</form>

</body>

</html>

53. resources\templates\blog\blog-edit.html

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">

<head>

<meta charset="UTF-8">

<title>Edit post</title>

</head>

<body>

<form th:action="${'/blog/edit/' +post.getId()}" th:object="${post}" method="post">

<input type="text" name="title" th:value="${post.title}" placeholder="Введите название статьи">

<input type="text" name="anons" th:value="${post.anons}" placeholder="Введите анос статьи">

<br>

<select th:field="\*{user}">

<option th:each="user : ${users}"

th:if="${(post.user != null && user.id == post.user.id) || user.post==null}"

th:value="${user.id}"

th:text="${user.username}">

</select>

<select th:field="\*{movie}">

<option th:each="element : ${movie}" th:text="${element.name}" th:value="${element.id}"></option>

</select>

<br>

<textarea name="full\_text" placeholder="Введите текст статьи"></textarea>

<br>

<button type="submit">Добавить статью</button>

</form>

</body>

</html>

54. resources\templates\blog\blog-filter.html

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">

<head>

<meta charset="UTF-8">

</head>

<body>

<div th:insert="header">...</div>

<h2>Фильтрация</h2>

<form method="post" th:action="${'/blog/filter/result'}">

<input type="text" name="title">

<input type="hidden"

th:name="${\_csrf.parameterName}"

th:value="${\_csrf.token}"/>

<button type="submit">Найти</button>

<div th:each="el : ${result}">

<h3 th:text="${el.title}"></h3>

<p th:text="${el.anons}"></p>

<strong th:text="${el.full\_text}"></strong>

</div>

</form>

</body>

</html>

55. resources\templates\blog\blog-main.html

<!DOCTYPE HTML>

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Getting Started: Serving Web Content</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

</head>

<body>

<div th:insert="header">...</div>

<h1>Все статьи</h1>

<div class="w-full max-w-xs">

<a href="/blog/filter" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Поиск

статей</a>

<a href="/blog/add" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Добавить

статью</a>

<div th:each="el : ${posts}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 class="font-bold" th:text="${el.title}"></h3>

<p th:text="${el.anons}"></p>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800" th:href="'/blog/edit/'+${el.id}">Редактировать</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/movie/del/'+${el.getId()}">Удалить</a>

<a th:href="'/blog/show/' + ${el.id}"

class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Подробнее</a>

</div>

</div>

</div>

</body>

</html>

56. resources\templates\blog\blog-show.html

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

<title th:value="${post.title}"></title>

</head>

<body>

<div th:insert="header">...</div>

<h3 th:text="${post.title}"></h3>

<p th:text="${post.getAnons()}"></p>

<br>

<p th:text="${post.full\_text}"></p>

<br>

<p th:text="${post.user.username}"></p>

<br>

<p th:text="${post.movie.name}"></p>

</body>

</html>

57. resources\templates\candidate\candidate-add.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<meta charset="UTF-8">

<title>Add candidate</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${candidate\_info}" th:action="${'/candidate/add'}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<label>

Дата загрузки:

<input th:field="\*{submissionDate}" type="date" name="submissionDate" placeholder="Введите дату загрузки"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('submissionDate')}" th:errors="\*{submissionDate}"></div>

<br>

<label> Отделение:

<select th:field="\*{target\_department}">

<option th:each="deps : ${departments}" th:text="${deps.name}" th:value="${deps.id}"></option>

</select>

</label> <br>

<label> Пользователь:

<select th:field="\*{user}">

<option th:each="usr : ${users}" th:text="${usr.login}" th:value="${usr.id}"></option>

</select>

</label> <br>

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Добавить кандидата

</button>

</form>

</div>

</body>

</html>

58. resources\templates\candidate\candidate-edit.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<head>

<meta charset="UTF-8">

<title>Edit preparation\_program</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${candidate\_info}" th:action="${'/candidate/edit/'+candidate\_info.id}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4"><label>

Дата загрузки:

<input th:field="\*{submissionDate}" type="date" name="submissionDate" placeholder="Введите дату загрузки"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('submissionDate')}" th:errors="\*{submissionDate}"></div>

<br>

<label> Отделение:

<select th:field="\*{target\_department}">

<option th:each="deps : ${departments}" th:text="${deps.name}" th:value="${deps.id}"></option>

</select>

</label> <br>

<input type="hidden" th:field="\*{user}" th:value="\*{user.id}">

<!-- <label> Пользователь:-->

<!-- <select th:field="\*{user}">-->

<!-- <option th:each="usr : ${users}" th:text="${usr.login}" th:value="${usr.id}"></option>-->

<!-- </select>-->

<!-- </label> <br>-->

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Редактировать документ

</button>

</form>

</div>

</body>

</html>

59. resources\templates\candidate\candidate-filter.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Filtered personal data</title>

<meta charset="UTF-8">

</head>

<body>

<div th:insert="header">...</div>

<h2>Фильтрация</h2>

<form>

<label>

<input type="text" name="title"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<input type="submit" formmethod="post" formaction="/document/filter/result" value="Найти"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

<input type="submit" formmethod="post" formaction="/document/filter\_strict/result" value="Найти точно"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

<input type="hidden"

th:name="${\_csrf.parameterName}"

th:value="${\_csrf.token}"/>

<div th:each="el : ${result}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 th:text="${el.user.login}" class="font-bold"></h3>

</div>

</div>

</form>

</body>

</html>

60. resources\templates\candidate\candidate-main.html

<!DOCTYPE HTML>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Candidate</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h1 class="font-bold">Все документы</h1>

<a href="/candidate/filter" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Поиск

документов</a>

<br>

<a href="/candidate/add" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Добавить

документ</a>

<div th:each="el : ${candidates}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 th:text="${el.submissionDate}" class="font-bold"></h3>

<h3 th:text="${el.user.login}"></h3>

<p>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/candidate/edit/'+${el.getId()}">Редактировать</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/candidate/show/'+${el.getId()}">Посмореть</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/candidate/del/'+${el.getId()}">Удалить</a>

</p>

</div>

</div>

</div>

</body>

</html>

61. resources\templates\candidate\candidate-show.html

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

<title th:value="${candidates.submissionDate}"></title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h3 class="font-bold" th:text="'Имя кандидата ' + ${candidates.user.login}"></h3>

<p th:text="'Дата поступления ' + ${candidates.submissionDate}"></p>

</div>

</body>

</html>

62. resources\templates\department\department-add.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<head>

<meta charset="UTF-8">

<title>Add department</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${department}" th:action="${'/department/add'}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<label>

Имя:

<input th:field="\*{name}" type="text" name="name" placeholder="Введите имя"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('name')}" th:errors="\*{name}"></div>

<br>

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Добавить отделение

</button>

</form>

</div>

</body>

</html>

63. resources\templates\department\department-edit.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<head>

<meta charset="UTF-8">

<title>Edit department</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${department}" th:action="${'/department/edit/'+department.getId()}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<label>

Имя:

<input th:field="\*{name}" type="text" name="name" placeholder="Введите имя"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('name')}" th:errors="\*{name}"></div>

<br>

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Редактировать отделение

</button>

</form>

</div>

</body>

</html>

64. resources\templates\department\department-filter.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Filtered personal data</title>

<meta charset="UTF-8">

</head>

<body>

<div th:insert="header">...</div>

<h2>Фильтрация</h2>

<form>

<label>

<input type="text" name="title"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<input type="submit" formmethod="post" formaction="/department/filter/result" value="Найти"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

<input type="submit" formmethod="post" formaction="/department/filter\_strict/result" value="Найти точно"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

<input type="hidden"

th:name="${\_csrf.parameterName}"

th:value="${\_csrf.token}"/>

<div th:each="el : ${result}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 th:text="${el.name}" class="font-bold"></h3>

</div>

</div>

</form>

</body>

</html>

65. resources\templates\department\department-main.html

<!DOCTYPE HTML>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Subject</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h1 class="font-bold">Все предметы</h1>

<a href="/department/filter" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Поиск

Отделений</a>

<br>

<a href="/department/add" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Добавить

Отделение</a>

<div th:each="el : ${department}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 th:text="${el.name}" class="font-bold"></h3>

<p>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/department/edit/'+${el.getId()}">Редактировать</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/department/show/'+${el.getId()}">Посмореть</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/department/del/'+${el.getId()}">Удалить</a>

</p>

</div>

</div>

</div>

</body>

</html>

66. resources\templates\department\department-show.html

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

<title th:value="${department.name}"></title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h3 class="font-bold" th:text="'Имя пользователя ' + ${department.getName()}"></h3>

</div>

</body>

</html>

67. resources\templates\document\document-add.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<meta charset="UTF-8">

<title>Add document</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${document}" th:action="${'/document/add'}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<label>

Имя:

<input th:field="\*{fileName}" type="text" name="fileName" placeholder="Введите имя"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('fileName')}" th:errors="\*{fileName}"></div>

<br>

<label>

Дата загрузки:

<input th:field="\*{date}" type="date" name="date" placeholder="Введите дату загрузки"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('date')}" th:errors="\*{date}"></div>

<br>

<label>

Дата архивации:

<input th:field="\*{archive\_date}" type="date" name="archive\_date" placeholder="Введите дату архивации"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('archive\_date')}" th:errors="\*{archive\_date}"></div>

<br>

<label> Пользователь:

<select th:field="\*{user}">

<option th:each="usr : ${users}" th:text="${usr.login}" th:value="${usr.id}"></option>

</select>

</label> <br>

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Добавить документ

</button>

</form>

</div>

</body>

</html>

68. resources\templates\document\document-edit.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<head>

<meta charset="UTF-8">

<title>Edit preparation\_program</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${documents}" th:action="${'/document/edit/'+documents.id}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<label>

Имя:

<input th:field="\*{fileName}" type="text" name="fileName" placeholder="Введите оценку"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('fileName')}" th:errors="\*{fileName}"></div>

<br>

<label>

Дата загрузки:

<input th:field="\*{date}" type="date" name="date" placeholder="Введите оценку"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('date')}" th:errors="\*{date}"></div>

<br>

<label>

Дата архивации:

<input th:field="\*{archive\_date}" type="date" name="archive\_date" placeholder="Введите оценку"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('archive\_date')}" th:errors="\*{archive\_date}"></div>

<br>

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Редактировать документ

</button>

</form>

</div>

</body>

</html>

69. resources\templates\document\document-filter.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Filtered personal data</title>

<meta charset="UTF-8">

</head>

<body>

<div th:insert="header">...</div>

<h2>Фильтрация</h2>

<form>

<label>

<input type="text" name="title"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<input type="submit" formmethod="post" formaction="/document/filter/result" value="Найти"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

<input type="submit" formmethod="post" formaction="/document/filter\_strict/result" value="Найти точно"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

<input type="hidden"

th:name="${\_csrf.parameterName}"

th:value="${\_csrf.token}"/>

<div th:each="el : ${result}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 th:text="${el.fileName}" class="font-bold"></h3>

</div>

</div>

</form>

</body>

</html>

70. resources\templates\document\document-main.html

<!DOCTYPE HTML>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Document</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h1 class="font-bold">Все документы</h1>

<a href="/document/filter" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Поиск

документов</a>

<br>

<a href="/document/add" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Добавить

документ</a>

<div th:each="el : ${documents}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 th:text="${el.getFileName()}" class="font-bold"></h3>

<h3 th:text="${el.getDate()}"></h3>

<h3 th:text="${el.user.login}"></h3>

<p>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/document/edit/'+${el.getId()}">Редактировать</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/document/show/'+${el.getId()}">Посмореть</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/document/del/'+${el.getId()}">Удалить</a>

</p>

</div>

</div>

</div>

</body>

</html>

71. resources\templates\document\document-show.html

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

<title th:value="${documents.fileName}"></title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h3 class="font-bold" th:text="'Имя файла ' + ${documents.fileName}"></h3>

<p th:text="'Имя пользователя ' + ${documents.user.login}"></p>

<p th:text="'Дата получения ' + ${documents.date}"></p>

<p th:text="'Дата архивации ' + ${documents.archive\_date}"></p>

</div>

</body>

</html>

72. resources\templates\education\education-add.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<head>

<meta charset="UTF-8">

<title>Add movie</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${education\_institution}" th:action="${'/education/add'}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<label>

Название:

<input th:field="\*{name}" type="text" name="name" placeholder="Введите название образовательного учреждения"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('name')}" th:errors="\*{name}"></div>

<br>

<label>

Адрес:

<input th:field="\*{address}" type="text" name="address" placeholder="Введите код предмета"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('address')}" th:errors="\*{address}"></div>

<br>

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Добавить образовательное учреждение

</button>

</form>

</div>

</body>

</html>

73. resources\templates\education\education-edit.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<head>

<meta charset="UTF-8">

<title> movie</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${education\_institution}" th:action="${'/education/edit/'+education\_institution.getId()}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<label>

Название:

<input th:field="\*{name}" type="text" name="name" placeholder="Введите название предмета"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('name')}" th:errors="\*{name}"></div>

<br>

<label>

Адрес:

<input th:field="\*{address}" type="text" name="address" placeholder="Введите код предмета"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('address')}" th:errors="\*{address}"></div>

<br>

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Редактировать образовательное учреждение

</button>

</form>

</div>

</body>

</html>

74. resources\templates\education\education-filter.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Filtered subjects</title>

<meta charset="UTF-8">

</head>

<body>

<div th:insert="header">...</div>

<h2>Фильтрация</h2>

<form>

<label>

<input type="text" name="title"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<input type="submit" formmethod="post" formaction="/education/filter/result" value="Найти"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

<input type="submit" formmethod="post" formaction="/education/filter\_strict/result" value="Найти точно"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

<input type="hidden"

th:name="${\_csrf.parameterName}"

th:value="${\_csrf.token}"/>

<div th:each="el : ${result}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 th:text="${el.name}" class="font-bold"></h3>

<p th:text="${el.address}"></p>

</div>

</div>

</form>

</body>

</html>

75. resources\templates\education\education-main.html

<!DOCTYPE HTML>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Subject</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h1 class="font-bold">Все предметы</h1>

<a href="/education/filter" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Поиск

об. Учеоеждений</a>

<br>

<a href="/education/add" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Добавить

об. Учеоеждение</a>

<div th:each="el : ${education\_institution}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 th:text="${el.name}" class="font-bold"></h3>

<p th:text="${el.address}"></p>

<p>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/education/edit/'+${el.getId()}">Редактировать</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/education/show/'+${el.getId()}">Посмореть</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/education/del/'+${el.getId()}">Удалить</a>

</p>

</div>

</div>

</div>

</body>

</html>

76. resources\templates\education\education-show.html

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

<title th:value="${education\_institution.name}"></title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h3 class="font-bold" th:text="'Об. Учреждение ' + ${education\_institution.getName()}"></h3>

<p th:text="${'Код предмета ' + education\_institution.getAddress()}"></p>

</div>

</body>

</html>

77. resources\templates\genre\genre-add.html

<!DOCTYPE HTML>

<html xmlns:th="http://www.thymeleaf.org" lang="ru">

<head>

<title>Добавление нового жанра</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:action="${'/genre/add'}" method="post" th:object="${genre}"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<div class="mb-4">

<label class="block text-gray-700 text-sm font-bold mb-2">Название</label>

<div style="color: red" th:if="${#fields.hasErrors('name')}" th:errors="\*{name}"></div>

<input th:field="\*{name}"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"

type="text" name="name" placeholder="Введите название жанра">

</div>

<div class="mb-4">

<label class="block text-gray-700 text-sm font-bold mb-2">Кино</label>

<div th:each="mov : ${movie}">

<input th:name="${'movie[]'}" th:value="${mov.id}" th:checked="${mov.id!=null && #lists.contains(mov.genres, mov)}" class="w-4 h-4 text-blue-600 bg-gray-100 rounded border-gray-300 focus:ring-blue-500 dark:focus:ring-blue-600 dark:ring-offset-gray-800 focus:ring-2 dark:bg-gray-700 dark:border-gray-600"

type="checkbox">

<label th:text="${mov.name}"></label>

</div>

<div style="color: red" th:if="${#fields.hasErrors('movie')}" th:errors="\*{movie}"></div>

</div>

<button class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline"

type="submit">Добавить жанр

</button>

</form>

</div>

</body>

</html>

78. resources\templates\genre\genre-edit.html

<!DOCTYPE HTML>

<html xmlns:th="http://www.thymeleaf.org" lang="ru">

<head>

<title th:value="${'Редактирование жанра'+genre.name}"></title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:action="${'/genre/edit/'+genre.id}" th:object="${genre}" method="post" class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<div class="mb-4">

<label class="block text-gray-700 text-sm font-bold mb-2">Название</label>

<div style="color: red" th:if="${#fields.hasErrors('name')}" th:errors="\*{name}"></div>

<input th:field="\*{name}"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"

type="text" name="name" placeholder="Введите название жанра">

</div>

<div class="mb-4">

<label class="block text-gray-700 text-sm font-bold mb-2">Кино</label>

<div th:each="mov : ${movie}">

<input th:name="${'movie[]'}" th:value="${mov.id}" th:checked="${mov.id!=null && #lists.contains(mov.genres, mov)}" class="w-4 h-4 text-blue-600 bg-gray-100 rounded border-gray-300 focus:ring-blue-500 dark:focus:ring-blue-600 dark:ring-offset-gray-800 focus:ring-2 dark:bg-gray-700 dark:border-gray-600"

type="checkbox">

<label th:text="${mov.name}"></label>

</div>

<input type="hidden" name="'movie[]" value="">

<div style="color: red" th:if="${#fields.hasErrors('movie')}" th:errors="\*{movie}"></div>

</div>

<button class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline"

type="submit">Сохранить

</button>

</form>

</div>

</body>

</html>

79. resources\templates\genre\genre-main.html

<!DOCTYPE HTML>

<html xmlns:th="http://www.thymeleaf.org" lang="ru">

<head>

<title>Жанры</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

</head>

<body>

<div th:insert="header">...</div>

<h1>Все Жанры</h1>

<a class="underline text-blue-600" href="/genre/add">Добавить жанр</a>

<div th:each="genre : ${genres}">

<div class="max-w-sm rounded overflow-hidden shadow-lg">

<div class="px-6 py-4">

<div class="font-bold text-xl mb-2" th:text="${genre.name}"/>

</div>

<div class="px-6 pt-4 pb-2">

<a class="underline text-blue-600" th:href="'/genre/edit/'+${genre.id}">Редактировать</a>

<a class="underline text-blue-600" th:href="'/genre/show/'+${genre.id}">Посмореть</a>

<a class="underline text-blue-600" th:href="'/genre/del/'+${genre.id}">Удалить</a>

</div>

</div>

</div>

</body>

</html>

80. resources\templates\genre\genre-show.html

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org" lang="ru">

<head>

<meta charset="UTF-8">

<title th:value="${genre.name}"></title>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

</head>

<body>

<div th:insert="header">...</div>

<div class="max-w-sm rounded overflow-hidden shadow-lg">

<div class="px-6 py-4">

<div class="font-bold text-xl mb-2" th:text="${genre.name}"/>

<div th:each="mov : ${genre.movie}">

<label th:text="${mov.name}" class="block text-gray-700 text-sm font-bold mb-2"></label>

</div>

</div>

</div>

</body>

</html>

81. resources\templates\group\group-add.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<head>

<meta charset="UTF-8">

<title>Add summary</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${group}" th:action="${'/group/add'}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<!-- th:checked="${groups!=null && #lists.contains(groups.getUsers(), user)}"-->

<div class="mb-4">

<label class="block text-gray-700 text-sm font-bold mb-2">Пользователи</label>

<div th:each="user : ${users}">

<input th:name="${'users[]'}" th:value="${user.id}" class="w-4 h-4 text-blue-600 bg-gray-100 rounded border-gray-300 focus:ring-blue-500 dark:focus:ring-blue-600 dark:ring-offset-gray-800 focus:ring-2 dark:bg-gray-700 dark:border-gray-600"

type="checkbox">

<label th:text="${user.login}"></label>

</div>

</div>

<label> Педагог огранизатор:

<select th:field="\*{teacher\_organizer}">

<option th:each="usr : ${users}" th:text="${usr.login}" th:value="${usr.id}"></option>

</select>

</label> <br>

<label> Программа подготовки:

<select th:field="\*{preparation\_program}">

<option th:each="preps : ${preparation\_programs}" th:text="${preps.name}" th:value="${preps.id}"></option>

</select>

</label> <br>

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Добавить группу

</button>

</form>

</div>

</body>

</html>

82. resources\templates\group\group-edit.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<head>

<meta charset="UTF-8">

<title>Edit preparation\_program</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${group}" th:action="${'/group/edit/'+group.getId()}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<div class="mb-4">

<label class="block text-gray-700 text-sm font-bold mb-2">Пользователи</label>

<div th:each="user : ${users}">

<input th:name="${'users[]'}" th:value="${user.id}" class="w-4 h-4 text-blue-600 bg-gray-100 rounded border-gray-300 focus:ring-blue-500 dark:focus:ring-blue-600 dark:ring-offset-gray-800 focus:ring-2 dark:bg-gray-700 dark:border-gray-600"

type="checkbox">

<label th:text="${user.login}"></label>

</div>

</div>

<label> Педагог огранизатор:

<select th:field="\*{teacher\_organizer}">

<option th:each="usr : ${users}" th:text="${usr.login}" th:value="${usr.id}"></option>

</select>

</label> <br>

<label> Программа подготовки:

<select th:field="\*{preparation\_program}">

<option th:each="preps : ${preparation\_programs}" th:text="${preps.name}" th:value="${preps.id}"></option>

</select>

</label> <br>

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Добавить отделение

</button>

</form>

</div>

</body>

</html>

83. resources\templates\group\group-filter.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Filtered personal data</title>

<meta charset="UTF-8">

</head>

<body>

<div th:insert="header">...</div>

<h2>Фильтрация</h2>

<form>

<label>

<input type="text" name="title"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<input type="submit" formmethod="post" formaction="/preparation\_program/filter/result" value="Найти"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

<input type="submit" formmethod="post" formaction="/preparation\_program/filter\_strict/result" value="Найти точно"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

<input type="hidden"

th:name="${\_csrf.parameterName}"

th:value="${\_csrf.token}"/>

<div th:each="el : ${result}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 th:text="${el.name}" class="font-bold"></h3>

</div>

</div>

</form>

</body>

</html>

84. resources\templates\group\group-main.html

<!DOCTYPE HTML>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Subject</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h1 class="font-bold">Все группы</h1>

<a href="/group/add" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Добавить

группу</a>

<div th:each="el : ${group}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 th:text="${el.teacher\_organizer.login}" class="font-bold"></h3>

<p th:text="${el.preparation\_program.name}"></p>

<p>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/group/edit/'+${el.getId()}">Редактировать</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/group/show/'+${el.getId()}">Посмореть</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/group/del/'+${el.getId()}">Удалить</a>

</p>

</div>

</div>

</div>

</body>

</html>

85. resources\templates\group\group-show.html

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

<title th:value="${groups.teacher\_organizer.login}"></title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h3 class="font-bold" th:text="'Имя педагога организатора ' + ${groups.teacher\_organizer.login}"></h3>

<h3 class="font-bold" th:text="'Название направления группы ' + ${groups.preparation\_program.name}"></h3>

<p>Ученики</p>

<p th:each="user: ${users}" th:text="${user.login}"></p>

</div>

</body>

</html>

86. resources\templates\header.html

<nav class="px-2 bg-white border-gray-200 dark:bg-gray-900 dark:border-gray-700"

xmlns:sec="http://www.w3.org/1999/xhtml">

<script src="https://unpkg.com/flowbite@1.5.3/dist/flowbite.js"></script>

<link rel="stylesheet" href="https://unpkg.com/flowbite@1.5.3/dist/flowbite.min.css"/>

<div class="container flex flex-wrap justify-between items-center mx-auto">

<button data-collapse-toggle="mobile-menu" type="button"

class="inline-flex justify-center items-center ml-3 text-gray-400 rounded-lg md:hidden hover:text-gray-900 focus:outline-none focus:ring-2 focus:ring-blue-300 dark:text-gray-400 dark:hover:text-white dark:focus:ring-gray-500"

aria-controls="mobile-menu-2" aria-expanded="false">

<span class="sr-only">Open main menu</span>

<svg class="w-6 h-6" aria-hidden="true" fill="currentColor" viewBox="0 0 20 20"

xmlns="http://www.w3.org/2000/svg">

<path fill-rule="evenodd"

d="M3 5a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1zM3 10a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1zM3 15a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1z"

clip-rule="evenodd"></path>

</svg>

</button>

<div class="flex items-center md:order-2">

<form sec:authorize="!isAuthenticated()" th:action="@{/login}" method="post">

<button type="submit" sec:authorize="!isAuthenticated()" href="/login"

class="text-gray-800 hover:bg-gray-50 focus:ring-4 focus:ring-gray-300 font-medium rounded-lg text-sm px-4 py-2 md:px-5 md:py-2.5 mr-1 md:mr-2 focus:outline-none">

Login

</button>

<a href="/registration"

class="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:ring-blue-300 font-medium rounded-lg text-sm px-4 py-2 md:px-5 md:py-2.5 mr-1 md:mr-2 focus:outline-none">Sign

up</a>

</form>

<form sec:authorize="isAuthenticated()" th:action="@{/logout}" method="post">

<button type="submit" href="/login"

class="text-gray-800 hover:bg-gray-50 focus:ring-4 focus:ring-gray-300 font-medium rounded-lg text-sm px-4 py-2 md:px-5 md:py-2.5 mr-1 md:mr-2 focus:outline-none">

Logout

</button>

</form>

</div>

<div class="hidden w-full md:block md:w-auto" id="mobile-menu">

<ul class="flex flex-col p-4 mt-4 bg-gray-50 rounded-lg border border-gray-100 md:flex-row md:space-x-8 md:mt-0 md:text-sm md:font-medium md:border-0 md:bg-white dark:bg-gray-800 md:dark:bg-gray-900 dark:border-gray-700">

<li>

<a href="/"

class="block py-2 pr-4 pl-3 text-white bg-blue-700 rounded md:bg-transparent md:text-blue-700 md:p-0 md:dark:text-white dark:bg-blue-600 md:dark:bg-transparent"

aria-current="page">Домашняя страница</a>

</li>

<li class="nav-item">

<a class="flex justify-between items-center py-2 pr-4 pl-3 w-full font-medium text-gray-700 rounded hover:bg-gray-100 md:hover:bg-transparent md:border-0 md:hover:text-blue-700 md:p-0 md:w-auto dark:text-gray-400 dark:hover:text-white dark:focus:text-white dark:border-gray-700 dark:hover:bg-gray-700 md:dark:hover:bg-transparent"

href="/blog" data-mdb-ripple="true" data-mdb-ripple-color="light">Блог</a>

</li>

<li class="nav-item">

<a class="flex justify-between items-center py-2 pr-4 pl-3 w-full font-medium text-gray-700 rounded hover:bg-gray-100 md:hover:bg-transparent md:border-0 md:hover:text-blue-700 md:p-0 md:w-auto dark:text-gray-400 dark:hover:text-white dark:focus:text-white dark:border-gray-700 dark:hover:bg-gray-700 md:dark:hover:bg-transparent"

href="/movie" data-mdb-ripple="true" data-mdb-ripple-color="light">Кино</a>

</li>

<li class="nav-item">

<a class="flex justify-between items-center py-2 pr-4 pl-3 w-full font-medium text-gray-700 rounded hover:bg-gray-100 md:hover:bg-transparent md:border-0 md:hover:text-blue-700 md:p-0 md:w-auto dark:text-gray-400 dark:hover:text-white dark:focus:text-white dark:border-gray-700 dark:hover:bg-gray-700 md:dark:hover:bg-transparent"

href="/genre" data-mdb-ripple="true" data-mdb-ripple-color="light">Жанр</a>

</li>

<li class="nav-item">

<a class="flex justify-between items-center py-2 pr-4 pl-3 w-full font-medium text-gray-700 rounded hover:bg-gray-100 md:hover:bg-transparent md:border-0 md:hover:text-blue-700 md:p-0 md:w-auto dark:text-gray-400 dark:hover:text-white dark:focus:text-white dark:border-gray-700 dark:hover:bg-gray-700 md:dark:hover:bg-transparent"

href="/document" data-mdb-ripple="true" data-mdb-ripple-color="light">Документ</a>

</li>

<li sec:authorize="hasAuthority('ADMIN')">

<button id="dropdownNavbarLink" data-dropdown-toggle="dropdownNavbar"

class="flex justify-between items-center py-2 pr-4 pl-3 w-full font-medium text-gray-700 rounded hover:bg-gray-100 md:hover:bg-transparent md:border-0 md:hover:text-blue-700 md:p-0 md:w-auto dark:text-gray-400 dark:hover:text-white dark:focus:text-white dark:border-gray-700 dark:hover:bg-gray-700 md:dark:hover:bg-transparent">

Меню администратора

<svg class="ml-1 w-5 h-5" aria-hidden="true" fill="currentColor" viewBox="0 0 20 20"

xmlns="http://www.w3.org/2000/svg">

<path fill-rule="evenodd"

d="M5.293 7.293a1 1 0 011.414 0L10 10.586l3.293-3.293a1 1 0 111.414 1.414l-4 4a1 1 0 01-1.414 0l-4-4a1 1 0 010-1.414z"

clip-rule="evenodd"></path>

</svg>

</button>

<!-- Dropdown menu -->

<div id="dropdownNavbar"

class="hidden z-10 w-44 font-normal bg-white rounded divide-y divide-gray-100 shadow dark:bg-gray-700 dark:divide-gray-600">

<ul class="py-1 text-sm text-gray-700 dark:text-gray-400"

aria-labelledby="dropdownLargeButton">

<li class="nav-item" sec:authorize="hasAuthority('ADMIN')">

<a class="block py-2 px-4 hover:bg-gray-100 dark:hover:bg-gray-600 dark:hover:text-white"

href="/user" data-mdb-ripple="true" data-mdb-ripple-color="light">Пользователи</a>

</li>

<li class="nav-item"

sec:authorize="hasAuthority('ADMIN') or hasAuthority('TEACHER')">

<a class="block py-2 px-4 hover:bg-gray-100 dark:hover:bg-gray-600 dark:hover:text-white"

href="/personal\_info" data-mdb-ripple="true" data-mdb-ripple-color="light">Пресональная

информация</a>

</li>

<li class="nav-item"

sec:authorize="hasAuthority('ADMIN') or hasAuthority('TEACHER')">

<a class="block py-2 px-4 hover:bg-gray-100 dark:hover:bg-gray-600 dark:hover:text-white"

href="/preparation\_program" data-mdb-ripple="true"

data-mdb-ripple-color="light">Программа

подготовки</a>

</li>

<li class="nav-item" sec:authorize="hasAuthority('ADMIN')">

<a class="block py-2 px-4 hover:bg-gray-100 dark:hover:bg-gray-600 dark:hover:text-white"

href="/subject" data-mdb-ripple="true"

data-mdb-ripple-color="light">Предмет</a>

</li>

<li class="nav-item"

sec:authorize="hasAuthority('ADMIN') or hasAuthority('TEACHER')">

<a class="block py-2 px-4 hover:bg-gray-100 dark:hover:bg-gray-600 dark:hover:text-white"

href="/department" data-mdb-ripple="true" data-mdb-ripple-color="light">Отделение</a>

</li>

<li class="nav-item"

sec:authorize="hasAuthority('ADMIN') or hasAuthority('TEACHER')">

<a class="block py-2 px-4 hover:bg-gray-100 dark:hover:bg-gray-600 dark:hover:text-white"

href="/education" data-mdb-ripple="true" data-mdb-ripple-color="light">Об.

учреждения</a>

</li>

<li class="nav-item"

sec:authorize="hasAuthority('ADMIN')">

<a class="block py-2 px-4 hover:bg-gray-100 dark:hover:bg-gray-600 dark:hover:text-white"

href="/group" data-mdb-ripple="true" data-mdb-ripple-color="light">Группа</a>

</li>

</ul>

</div>

</li>

<li sec:authorize="hasAuthority('TEACHER')">

<button id="teachDropdownNavbarLink" data-dropdown-toggle="teachDropdownNavbar"

class="flex justify-between items-center py-2 pr-4 pl-3 w-full font-medium text-gray-700 rounded hover:bg-gray-100 md:hover:bg-transparent md:border-0 md:hover:text-blue-700 md:p-0 md:w-auto dark:text-gray-400 dark:hover:text-white dark:focus:text-white dark:border-gray-700 dark:hover:bg-gray-700 md:dark:hover:bg-transparent">

Меню препода

<svg class="ml-1 w-5 h-5" aria-hidden="true" fill="currentColor" viewBox="0 0 20 20"

xmlns="http://www.w3.org/2000/svg">

<path fill-rule="evenodd"

d="M5.293 7.293a1 1 0 011.414 0L10 10.586l3.293-3.293a1 1 0 111.414 1.414l-4 4a1 1 0 01-1.414 0l-4-4a1 1 0 010-1.414z"

clip-rule="evenodd"></path>

</svg>

</button>

<!-- Dropdown menu -->

<div id="teachDropdownNavbar"

class="hidden z-10 w-44 font-normal bg-white rounded divide-y divide-gray-100 shadow dark:bg-gray-700 dark:divide-gray-600">

<ul class="py-1 text-sm text-gray-700 dark:text-gray-400"

aria-labelledby="dropdownLargeButton">

<li class="nav-item"

sec:authorize="hasAuthority('ADMIN') or hasAuthority('TEACHER')">

<a class="block py-2 px-4 hover:bg-gray-100 dark:hover:bg-gray-600 dark:hover:text-white"

href="/department" data-mdb-ripple="true" data-mdb-ripple-color="light">Отделение</a>

</li>

<li class="nav-item"

sec:authorize="hasAuthority('ADMIN') or hasAuthority('TEACHER')">

<a class="block py-2 px-4 hover:bg-gray-100 dark:hover:bg-gray-600 dark:hover:text-white"

href="/education" data-mdb-ripple="true" data-mdb-ripple-color="light">Об.

учреждения</a>

</li>

<li class="nav-item"

sec:authorize="hasAuthority('ADMIN') or hasAuthority('TEACHER')">

<a class="block py-2 px-4 hover:bg-gray-100 dark:hover:bg-gray-600 dark:hover:text-white"

href="/department" data-mdb-ripple="true" data-mdb-ripple-color="light">Отделение</a>

</li>

<li class="nav-item"

sec:authorize="hasAuthority('ADMIN') or hasAuthority('TEACHER')">

<a class="block py-2 px-4 hover:bg-gray-100 dark:hover:bg-gray-600 dark:hover:text-white"

href="/preparation\_program" data-mdb-ripple="true"

data-mdb-ripple-color="light">Программа

подготовки</a>

</li>

<li class="nav-item"

sec:authorize="hasAuthority('ADMIN') or hasAuthority('TEACHER')">

<a class="block py-2 px-4 hover:bg-gray-100 dark:hover:bg-gray-600 dark:hover:text-white"

href="/personal\_info" data-mdb-ripple="true" data-mdb-ripple-color="light">Пресональная

информация</a>

</li>

</ul>

</div>

</li>

<li sec:authorize="hasAuthority('TEACHER') or hasAuthority('USER')">

<button id="userDropdownNavbarLink" data-dropdown-toggle="userDropdownNavbar"

class="flex justify-between items-center py-2 pr-4 pl-3 w-full font-medium text-gray-700 rounded hover:bg-gray-100 md:hover:bg-transparent md:border-0 md:hover:text-blue-700 md:p-0 md:w-auto dark:text-gray-400 dark:hover:text-white dark:focus:text-white dark:border-gray-700 dark:hover:bg-gray-700 md:dark:hover:bg-transparent">

Меню пользователя

<svg class="ml-1 w-5 h-5" aria-hidden="true" fill="currentColor" viewBox="0 0 20 20"

xmlns="http://www.w3.org/2000/svg">

<path fill-rule="evenodd"

d="M5.293 7.293a1 1 0 011.414 0L10 10.586l3.293-3.293a1 1 0 111.414 1.414l-4 4a1 1 0 01-1.414 0l-4-4a1 1 0 010-1.414z"

clip-rule="evenodd"></path>

</svg>

</button>

<!-- Dropdown menu -->

<div id="userDropdownNavbar"

class="hidden z-10 w-44 font-normal bg-white rounded divide-y divide-gray-100 shadow dark:bg-gray-700 dark:divide-gray-600">

<ul class="py-1 text-sm text-gray-700 dark:text-gray-400"

aria-labelledby="dropdownLargeButton">

<li class="nav-item"

sec:authorize="hasAuthority('TEACHER') or hasAuthority('User')">

<a class="block py-2 px-4 hover:bg-gray-100 dark:hover:bg-gray-600 dark:hover:text-white"

href="/summary" data-mdb-ripple="true" data-mdb-ripple-color="light">Информаиция о

пользователе</a>

</li>

</ul>

</div>

</li>

</ul>

</div>

</div>

</nav>

87. resources\templates\home.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Главная</title>

</head>

<body>

<div th:insert="header">...</div>

<div style="width: 50%" class="tenor-gif-embed" data-postid="18438923" data-share-method="host" data-aspect-ratio="1.77778" data-width="100%"><a href="https://tenor.com/view/jrr-gif-18438923">Jrr GIF</a>from <a href="https://tenor.com/search/jrr-gifs">Jrr GIFs</a></div> <script type="text/javascript" async src="https://tenor.com/embed.js"></script>

</body>

</html>

88. resources\templates\login.html

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="https://www.thymeleaf.org"

xmlns:sec="https://www.thymeleaf.org/thymeleaf-extras-springsecurity3">

<head>

<title>Login</title>

</head>

<body>

<div th:if="${param.error}">

Invalid username and password.

</div>

<div th:if="${param.logout}">

You have been logged out.

</div>

<form th:action="@{/login}" method="post">

<div><label> User Name : <input type="text" name="login"/> </label></div>

<div><label> Password: <input type="password" name="password"/> </label></div>

<div><input type="submit" value="Log In"/></div>

</form>

<a href="/registration">Регистрация нового пользователя</a>

</body>

</html>

89. resources\templates\movie\movie-add.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<head>

<meta charset="UTF-8">

<title>Add movie</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${movie}" th:action="${'/movie/add'}" method="post" class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<label>

Название:

<input th:field="\*{name}" type="text" name="name" placeholder="Введите название кино"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('name')}" th:errors="\*{name}"></div>

<br>

<label>

Дата выпуска:

<input th:field="\*{release\_date}" type="date" name="release\_date" placeholder="Введите дату выпуска"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"/>

</label>

<div style="color: red" th:if="${#fields.hasErrors('release\_date')}" th:errors="\*{release\_date}"></div>

<br>

<label>

Рейтинг:

<input th:field="\*{rating}" type="number" name="rating" placeholder="Введите рейтинг"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"/>

</label>

<div style="color: red" th:if="${#fields.hasErrors('rating')}" th:errors="\*{rating}"></div>

<br>

<label>

Количество комметариев:

<input th:field="\*{series\_amount}" type="number" name="series\_amount" placeholder="Введите количество комметариев"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"

/>

</label>

<div style="color: red" th:if="${#fields.hasErrors('series\_amount')}" th:errors="\*{series\_amount}"></div>

<br>

<label>

Количество серий:

<input th:field="\*{comments\_amount}" type="number" name="comments\_amount" placeholder="Введите кол-во просмотров"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"/>

</label>

<div style="color: red" th:if="${#fields.hasErrors('comments\_amount')}" th:errors="\*{comments\_amount}"></div>

<br>

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Добавить кино/сериал

</button>

</form>

</div>

</body>

</html>

90. resources\templates\movie\movie-edit.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html lang="en">

<head>

<title>Movie</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${movie}" th:action="${'/movie/edit/'+movie.getId()}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<label>

Название:

<input th:field="\*{name}" type="text" name="name" placeholder="Введите название кино"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('name')}" th:errors="\*{name}"></div>

<br>

<label>

Дата выпуска:

<input th:field="\*{release\_date}" type="date" name="release\_date" placeholder="Введите дату выпуска"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"/>

</label>

<div style="color: red" th:if="${#fields.hasErrors('release\_date')}" th:errors="\*{release\_date}"></div>

<br>

<label>

Рейтинг:

<input th:field="\*{rating}" type="number" name="rating" placeholder="Введите рейтинг"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"/>

</label>

<div style="color: red" th:if="${#fields.hasErrors('rating')}" th:errors="\*{rating}"></div>

<br>

<label>

Количество комметариев:

<input th:field="\*{series\_amount}" type="number" name="series\_amount" placeholder="Введите количество комметариев"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"

/>

</label>

<div style="color: red" th:if="${#fields.hasErrors('series\_amount')}" th:errors="\*{series\_amount}"></div>

<br>

<label>

Количество серий:

<input th:field="\*{comments\_amount}" type="number" name="comments\_amount" placeholder="Введите кол-во просмотров"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"/>

</label>

<div style="color: red" th:if="${#fields.hasErrors('comments\_amount')}" th:errors="\*{comments\_amount}"></div>

<br>

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Добавить кино/сериал

</button>

</form>

</div>

</body>

</html>

91. resources\templates\movie\movie-filter.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Filtered movies</title>

<meta charset="UTF-8">

</head>

<body>

<div th:insert="header">...</div>

<h2>Фильтрация</h2>

<form>

<label>

<input type="text" name="title">

</label>

<input type="submit" formmethod="post" formaction="/movie/filter/result" value="Найти">

<input type="submit" formmethod="post" formaction="/movie/filter\_strict/result" value="Найти точно">

<input type="hidden"

th:name="${\_csrf.parameterName}"

th:value="${\_csrf.token}"/>

<div th:each="el : ${result}">

<h3 th:text="${el.name}"></h3>

<p th:text="${el.release\_date}"></p>

</div>

</form>

</body>

</html>

92. resources\templates\movie\movie-main.html

<!DOCTYPE HTML>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Movie</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h1 class="font-bold">Все кино/сериалы</h1>

<a href="/movie/filter" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Поиск

кино/сериалов</a>

<br>

<a href="/movie/add" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Добавить

кино/сериал</a>

<div th:each="el : ${movie}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 th:text="${el.name}" class="font-bold"></h3>

<p th:text="${el.release\_date}"></p>

<p>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/movie/edit/'+${el.getId()}">Редактировать</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/movie/show/'+${el.getId()}">Посмореть</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/movie/del/'+${el.getId()}">Удалить</a>

</p>

</div>

</div>

</div>

</body>

</html>

93. resources\templates\movie\movie-show.html

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

<title th:value="${movie.name}"></title>

</head>

<body>

<div th:insert="header">...</div>

<h3 th:text="${movie.getName()}"></h3>

<label>

Количество серий

<p th:text="${movie.getSeries\_amount()}"></p>

</label>

<p th:text="${movie.getComments\_amount()}"></p>

<p th:text="${movie.getRating()}"></p>

<p th:text="${movie.getRelease\_date()}"></p>

</body>

</html>

94. resources\templates\personal\_info\personal\_info-add.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<head>

<meta charset="UTF-8">

<title>Add movie</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${personal\_info}" th:action="${'/personal\_info/add'}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<label>

Имя:

<input th:field="\*{name}" type="text" name="name" placeholder="Введите имя"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('name')}" th:errors="\*{name}"></div>

<br>

<label>

Фамилия:

<input th:field="\*{surname}" type="text" name="surname" placeholder="Введите фамилию"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('surname')}" th:errors="\*{surname}"></div>

<br>

<label>

Отчество:

<input th:field="\*{sec\_name}" type="text" name="sec\_name" placeholder="Введите отчество"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('sec\_name')}" th:errors="\*{sec\_name}"></div>

<br>

<label>

Дата рождения

<input required th:field="\*{birthdate}" type="date" name="birthdate" placeholder="Введите дату рождения"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"/>

</label>

<div style="color: red" th:if="${#fields.hasErrors('birthdate')}" th:errors="\*{birthdate}"></div>

<br>

<label>

Серия паспорта:

<input th:field="\*{pass\_id}" type="text" name="pass\_id" placeholder="Введите серию паспорта"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('pass\_id')}" th:errors="\*{pass\_id}"></div>

<br>

<label>

Является ли мужчиной:

<input th:field="\*{is\_male}" type="checkbox" name="is\_male">

</label>

<div style="color: red" th:if="${#fields.hasErrors('is\_male')}" th:errors="\*{pass\_id}"></div>

<br>

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Добавить персональные данные

</button>

</form>

</div>

</body>

</html>

95. resources\templates\personal\_info\personal\_info-edit.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<head>

<meta charset="UTF-8">

<title>personal\_info</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${personal\_info}" th:action="${'/personal\_info/edit/'+personal\_info.getId()}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<label>

Имя:

<input th:field="\*{name}" type="text" name="name" placeholder="Введите имя"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('name')}" th:errors="\*{name}"></div>

<br>

<label>

Фамилия:

<input th:field="\*{surname}" type="text" name="surname" placeholder="Введите фамилию"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('surname')}" th:errors="\*{surname}"></div>

<br>

<label>

Отчество:

<input th:field="\*{sec\_name}" type="text" name="sec\_name" placeholder="Введите отчество"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('sec\_name')}" th:errors="\*{sec\_name}"></div>

<br>

<label>

Дата рождения

<input required th:field="\*{birthdate}" type="date" name="birthdate" placeholder="Введите дату рождения"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"/>

</label>

<div style="color: red" th:if="${#fields.hasErrors('birthdate')}" th:errors="\*{birthdate}"></div>

<br>

<label>

Серия паспорта:

<input th:field="\*{pass\_id}" type="text" name="pass\_id" placeholder="Введите серию паспорта"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('pass\_id')}" th:errors="\*{pass\_id}"></div>

<br>

<label>

Является ли мужчиной:

<input th:field="\*{is\_male}" type="checkbox" name="is\_male">

</label>

<div style="color: red" th:if="${#fields.hasErrors('is\_male')}" th:errors="\*{pass\_id}"></div>

<br>

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Редактировать персональные данные

</button>

</form>

</div>

</body>

</html>

96. resources\templates\personal\_info\personal\_info-filter.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Filtered personal data</title>

<meta charset="UTF-8">

</head>

<body>

<div th:insert="header">...</div>

<h2>Фильтрация</h2>

<form>

<label>

<input type="text" name="title"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<input type="submit" formmethod="post" formaction="/personal\_info/filter/result" value="Найти"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

<input type="submit" formmethod="post" formaction="/personal\_info/filter\_strict/result" value="Найти точно"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

<input type="hidden"

th:name="${\_csrf.parameterName}"

th:value="${\_csrf.token}"/>

<div th:each="el : ${result}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 th:text="${el.name}" class="font-bold"></h3>

<p th:text="${el.surname}"></p>

<p th:text="${el.sec\_name}"></p>

<p th:text="${el.birthdate}"></p>

<p th:text="${el.pass\_id}"></p>

</div>

</div>

</form>

</body>

</html>

97. resources\templates\personal\_info\personal\_info-main.html

<!DOCTYPE HTML>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Subject</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h1 class="font-bold">Все предметы</h1>

<a href="/personal\_info/filter" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Поиск

Пресональных данных</a>

<br>

<a href="/personal\_info/add" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Добавить

Пресональные данные</a>

<div th:each="el : ${personal\_info}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 th:text="${el.name}" class="font-bold"></h3>

<p th:text="${el.surname}"></p>

<p th:text="${el.sec\_name}"></p>

<p>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/personal\_info/edit/'+${el.getId()}">Редактировать</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/personal\_info/show/'+${el.getId()}">Посмореть</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/personal\_info/del/'+${el.getId()}">Удалить</a>

</p>

</div>

</div>

</div>

</body>

</html>

98. resources\templates\personal\_info\personal\_info-show.html

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

<title th:value="${personal\_info.name}"></title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h3 class="font-bold" th:text="'Об. Учреждение ' + ${personal\_info.getName()}"></h3>

<p th:text="${'Код предмета ' + personal\_info.getAddress()}"></p>

<p th:text="${personal\_info.surname}"></p>

<p th:text="${personal\_info.sec\_name}"></p>

<p th:text="${personal\_info.birthdate}"></p>

<p th:text="${personal\_info.pass\_id}"></p>

</div>

</body>

</html>

99. resources\templates\preparation\_program\preparation\_program-add.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<head>

<meta charset="UTF-8">

<title>Add preparation\_program</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${preparation\_program}" th:action="${'/preparation\_program/add'}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<label>

Имя:

<input th:field="\*{name}" type="text" name="name" placeholder="Введите имя"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('name')}" th:errors="\*{name}"></div>

<br>

<div class="mb-4">

<label>Предметы</label>

<div th:each="sub : ${subjectSet}">

<input th:name="${'subjectSet[]'}" th:value="${sub.id}"

th:checked="${sub.id!=null && #lists.contains(sub.preparation\_programs, sub)}"

class="w-4 h-4 text-blue-600 bg-gray-100 rounded border-gray-300 focus:ring-blue-500 dark:focus:ring-blue-600 dark:ring-offset-gray-800 focus:ring-2 dark:bg-gray-700 dark:border-gray-600"

type="checkbox">

<label th:text="${sub.name}"></label>

</div>

</div>

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Добавить отделение

</button>

</form>

</div>

</body>

</html>

100. resources\templates\preparation\_program\preparation\_program-edit.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<head>

<meta charset="UTF-8">

<title>Edit preparation\_program</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${preparation\_program}" th:action="${'/preparation\_program/edit/'+preparation\_program.getId()}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<label>

Имя:

<input th:field="\*{name}" type="text" name="name" placeholder="Введите имя"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('name')}" th:errors="\*{name}"></div>

<br>

<div class="mb-4">

<label>Предметы</label>

<div th:each="sub : ${subjectSet}">

<input th:name="${'subjectSet[]'}" th:value="${sub.id}"

th:checked="${sub.id!=null && #lists.contains(sub.preparation\_programs, preparation\_program)}"

class="w-4 h-4 text-blue-600 bg-gray-100 rounded border-gray-300 focus:ring-blue-500 dark:focus:ring-blue-600 dark:ring-offset-gray-800 focus:ring-2 dark:bg-gray-700 dark:border-gray-600"

type="checkbox">

<label th:text="${sub.name}"></label>

</div>

</div>

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Редактировать отделение

</button>

</form>

</div>

</body>

</html>

101. resources\templates\preparation\_program\preparation\_program-filter.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Filtered personal data</title>

<meta charset="UTF-8">

</head>

<body>

<div th:insert="header">...</div>

<h2>Фильтрация</h2>

<form>

<label>

<input type="text" name="title"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<input type="submit" formmethod="post" formaction="/preparation\_program/filter/result" value="Найти"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

<input type="submit" formmethod="post" formaction="/preparation\_program/filter\_strict/result" value="Найти точно"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

<input type="hidden"

th:name="${\_csrf.parameterName}"

th:value="${\_csrf.token}"/>

<div th:each="el : ${result}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 th:text="${el.name}" class="font-bold"></h3>

</div>

</div>

</form>

</body>

</html>

102. resources\templates\preparation\_program\preparation\_program-main.html

<!DOCTYPE HTML>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Subject</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h1 class="font-bold">Все предметы</h1>

<a href="/preparation\_program/filter" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Поиск

Программ подготовок</a>

<br>

<a href="/preparation\_program/add" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Добавить

Программу подготовки</a>

<div th:each="el : ${preparation\_program}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 th:text="${el.name}" class="font-bold"></h3>

<p>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/preparation\_program/edit/'+${el.getId()}">Редактировать</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/preparation\_program/show/'+${el.getId()}">Посмореть</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/preparation\_program/del/'+${el.getId()}">Удалить</a>

</p>

</div>

</div>

</div>

</body>

</html>

103. resources\templates\preparation\_program\preparation\_program-show.html

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

<title th:value="${preparation\_program.name}"></title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h3 class="font-bold" th:text="'Имя пользователя ' + ${preparation\_program.getName()}"></h3>

</div>

</body>

</html>

104. resources\templates\registration.html

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="https://www.thymeleaf.org"

xmlns:sec="https://www.thymeleaf.org/thymeleaf-extras-springsecurity3">

<head>

<title>Register</title>

</head>

<body>

<div th:if="${#messages}">

<p th:text="${message}"></p>

</div>

Регистрация нового пользователя

<form th:action="@{/registration}" method="post">

<div><label> User Name : <input type="text" name="login"/> </label></div>

<div><label> Password: <input type="password" name="password"/> </label></div>

<div><input type="submit" value="Reg"/></div>

</form>

</body>

</html>

105. resources\templates\subject\subject-add.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<head>

<meta charset="UTF-8">

<title>Add movie</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${subject}" th:action="${'/subject/add'}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<label>

Название:

<input th:field="\*{name}" type="text" name="name" placeholder="Введите название предмета"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('name')}" th:errors="\*{name}"></div>

<br>

<label>

Код предмета:

<input th:field="\*{subject\_code}" type="text" name="subject\_code" placeholder="Введите код предмета"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('subject\_code')}" th:errors="\*{subject\_code}"></div>

<br>

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Добавить предмет

</button>

</form>

</div>

</body>

</html>

106. resources\templates\subject\subject-edit.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<head>

<meta charset="UTF-8">

<title> movie</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${subject}" th:action="${'/subject/edit/'+subject.getId()}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<label>

Название:

<input th:field="\*{name}" type="text" name="name" placeholder="Введите название предмета"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('name')}" th:errors="\*{name}"></div>

<br>

<label>

Код предмета:

<input th:field="\*{subject\_code}" type="text" name="subject\_code" placeholder="Введите код предмета"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('subject\_code')}" th:errors="\*{subject\_code}"></div>

<br>

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Редактировать предмет

</button>

</form>

</div>

</body>

</html>

107. resources\templates\subject\subject-filter.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Filtered subjects</title>

<meta charset="UTF-8">

</head>

<body>

<div th:insert="header">...</div>

<h2>Фильтрация</h2>

<form>

<label>

<input type="text" name="title"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<input type="submit" formmethod="post" formaction="/subject/filter/result" value="Найти"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

<input type="submit" formmethod="post" formaction="/subject/filter\_strict/result" value="Найти точно"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

<input type="hidden"

th:name="${\_csrf.parameterName}"

th:value="${\_csrf.token}"/>

<div th:each="el : ${result}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 th:text="${el.name}" class="font-bold"></h3>

<p th:text="${el.subject\_code}"></p>

</div>

</div>

</form>

</body>

</html>

108. resources\templates\subject\subject-main.html

<!DOCTYPE HTML>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Subject</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h1 class="font-bold">Все предметы</h1>

<a href="/subject/filter" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Поиск

предметов</a>

<br>

<a href="/subject/add" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Добавить

предмет</a>

<div th:each="el : ${subject}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 th:text="${el.name}" class="font-bold"></h3>

<p th:text="${el.subject\_code}"></p>

<p>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/subject/edit/'+${el.getId()}">Редактировать</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/subject/show/'+${el.getId()}">Посмореть</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/subject/del/'+${el.getId()}">Удалить</a>

</p>

</div>

</div>

</div>

</body>

</html>

109. resources\templates\subject\subject-show.html

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

<title th:value="${subject.name}"></title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h3 class="font-bold" th:text="Предмет + ${subject.getName()}"></h3>

<p th:text="Код предмета + ${subject.getSubject\_code()}"></p>

</div>

</body>

</html>

110. resources\templates\summary\summary-add.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<head>

<meta charset="UTF-8">

<title>Add summary</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${summary}" th:action="${'/summary/add'}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<label>

Имя:

<input th:field="\*{mark}" type="number" name="mark" placeholder="Введите оценку"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('mark')}" th:errors="\*{mark}"></div>

<br>

<label> Образовательное учреждение:

<select th:field="\*{education\_institution}">

<option th:each="edu : ${education\_institution}" th:text="${edu.getName()}" th:value="${edu.getName()}"></option>

</select>

</label> <br>

<label> Предмет:

<select th:field="\*{subject}">

<option th:each="sub : ${subjects}" th:text="${sub.name}" th:value="${sub.name}"></option>

</select>

</label>

<br>

<label> Кандидат:

<select th:field="\*{candidate\_info}">

<option th:each="can : ${candidate}" th:text="${can.getUser().login}" th:value="${can.getId()}"></option>

</select>

</label> <br>

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Добавить отделение

</button>

</form>

</div>

</body>

</html>

111. resources\templates\summary\summary-edit.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<head>

<meta charset="UTF-8">

<title>Edit preparation\_program</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${preparation\_program}" th:action="${'/preparation\_program/edit/'+preparation\_program.getId()}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<label>

Имя:

<input th:field="\*{name}" type="text" name="name" placeholder="Введите имя"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<div style="color: red" th:if="${#fields.hasErrors('name')}" th:errors="\*{name}"></div>

<br>

<div class="mb-4">

<label>Предметы</label>

<div th:each="sub : ${subjectSet}">

<input th:name="${'subjectSet[]'}" th:value="${sub.id}"

th:checked="${sub.id!=null && #lists.contains(sub.preparation\_programs, preparation\_program)}"

class="w-4 h-4 text-blue-600 bg-gray-100 rounded border-gray-300 focus:ring-blue-500 dark:focus:ring-blue-600 dark:ring-offset-gray-800 focus:ring-2 dark:bg-gray-700 dark:border-gray-600"

type="checkbox">

<label th:text="${sub.name}"></label>

</div>

</div>

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Редактировать отделение

</button>

</form>

</div>

</body>

</html>

112. resources\templates\summary\summary-filter.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Filtered personal data</title>

<meta charset="UTF-8">

</head>

<body>

<div th:insert="header">...</div>

<h2>Фильтрация</h2>

<form>

<label>

<input type="text" name="title"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline">

</label>

<input type="submit" formmethod="post" formaction="/preparation\_program/filter/result" value="Найти"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

<input type="submit" formmethod="post" formaction="/preparation\_program/filter\_strict/result" value="Найти точно"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

<input type="hidden"

th:name="${\_csrf.parameterName}"

th:value="${\_csrf.token}"/>

<div th:each="el : ${result}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 th:text="${el.name}" class="font-bold"></h3>

</div>

</div>

</form>

</body>

</html>

113. resources\templates\summary\summary-main.html

<!DOCTYPE HTML>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Subject</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h1 class="font-bold">Все предметы</h1>

<a href="/summary/filter" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Поиск

Программ подготовок</a>

<br>

<a href="/summary/add" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Добавить

Программу подготовки</a>

<div th:each="el : ${summary}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 th:text="${el.mark}" class="font-bold"></h3>

<p th:text="${el.getCandidate\_info().getUser\_candidate\_info().name}" class="font-bold"></p>

<p>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/summary/edit/'+${el.getId()}">Редактировать</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/summary/show/'+${el.getId()}">Посмореть</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/summary/del/'+${el.getId()}">Удалить</a>

</p>

</div>

</div>

</div>

</body>

</html>

114. resources\templates\summary\summary-show.html

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

<title th:value="${summary.name}"></title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h3 class="font-bold" th:text="'Имя пользователя ' + ${summary.getName()}"></h3>

</div>

</body>

</html>

115. resources\templates\teacher\teacher-add.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<head>

<meta charset="UTF-8">

<title>Add summary</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${teacher}" th:action="${'/teacher/add'}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<div class="mb-4">

<label class="block text-gray-700 text-sm font-bold mb-2">Пользователи</label>

<div th:each="sub : ${subjects}">

<input th:name="${'subjectList[]'}" th:value="${sub.id}" th:checked="${sub.id!=null && #lists.contains(sub.teacher\_infoList, teacher)}" class="w-4 h-4 text-blue-600 bg-gray-100 rounded border-gray-300 focus:ring-blue-500 dark:focus:ring-blue-600 dark:ring-offset-gray-800 focus:ring-2 dark:bg-gray-700 dark:border-gray-600"

type="checkbox">

<label th:text="${sub.name}"></label>

</div>

</div>

<!-- <label >-->

<!-- Стаж работы с ...-->

<!-- <input required th:field="\*{teachSince}" type="date" name="teachSince" placeholder="Введите cтаж работы"-->

<!-- class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"/>-->

<!-- </label>-->

<!--&lt;!&ndash; <div style="color: red" th:if="${#fields.hasErrors('teachSince')}" th:errors="\*{teachSince}"></div>&ndash;&gt;-->

<!-- <br>-->

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Добавить иформацию о преподе

</button>

</form>

</div>

</body>

</html>

116. resources\templates\teacher\teacher-edit.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<head>

<meta charset="UTF-8">

<title>Edit preparation\_program</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${teacher}" th:action="${'/teacher/edit/'+teacher.getId()}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<div class="mb-4">

<label class="block text-gray-700 text-sm font-bold mb-2">Пользователи</label>

<div th:each="sub : ${subjects}">

<input th:name="${'subject[]'}" th:value="${sub.id}" th:checked="${sub.id!=null && #lists.contains(sub.teacher\_infoList, teacher)}" class="w-4 h-4 text-blue-600 bg-gray-100 rounded border-gray-300 focus:ring-blue-500 dark:focus:ring-blue-600 dark:ring-offset-gray-800 focus:ring-2 dark:bg-gray-700 dark:border-gray-600"

type="checkbox">

<label th:text="${sub.name}"></label>

</div>

</div>

<!-- <label >-->

<!-- Стаж работы с ...-->

<!-- <input required th:field="\*{teachSince}" type="date" name="teachSince" placeholder="Введите cтаж работы"-->

<!-- class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"/>-->

<!-- </label>-->

<!-- <div style="color: red" th:if="${#fields.hasErrors('teachSince')}" th:errors="\*{teachSince}"></div>-->

<!-- <br>-->

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Изменить иформацию о преподе

</button>

</form>

</div>

</body>

</html>

117. resources\templates\teacher\teacher-main.html

<!DOCTYPE HTML>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Teacher info</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h1 class="font-bold">Все преподы</h1>

<a href="/teacher/add" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Добавить

информацию о преподу/a>

<div th:each="el : ${teacher}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 th:text="${el.teachSince}" class="font-bold"></h3>

<p>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/teacher/edit/'+${el.getId()}">Редактировать</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/teacher/show/'+${el.getId()}">Посмореть</a>

<a class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800"

th:href="'/teacher/del/'+${el.getId()}">Удалить</a>

</p>

</div>

</div>

</div>

</body>

</html>

118. resources\templates\teacher\teacher-show.html

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

<title th:value="${groups.teacher\_organizer.login}"></title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h3 class="font-bold" th:text="'Имя педагога организатора ' + ${groups.teacher\_organizer.login}"></h3>

<p>Ученики</p>

<p th:each="user: ${users}" th:text="${user.login}"></p>

</div>

</body>

</html>

119. resources\templates\user\user-add.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<meta charset="UTF-8">

<title>Add user</title>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${user}" th:action="${'/user/add'}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<label >

Введите имя

<input required th:field="\*{login}" type="text" name="login" placeholder="Имя"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"/>

</label>

<div style="color: red" th:if="${#fields.hasErrors('login')}" th:errors="\*{login}"></div>

<br>

<label >

Введите пароль

<input required th:field="\*{password}" type="password" name="password" placeholder="Имя"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"/>

</label>

<div style="color: red" th:if="${#fields.hasErrors('password')}" th:errors="\*{password}"></div>

<br>

<br>

<div th:each="role : ${roles}">

<label>

<input th:id="${role}" type="checkbox" th:name="${'roles[]'}" th:value="${role}"

/>

</label>

<label th:text="${role}"/>

<br>

</div>

<div class="flex items-center justify-between">

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Добавить пользователя

</button>

</div>

</form>

</div>

</body>

</html>

120. resources\templates\user\user-edit.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<head>

<title>Movie</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form th:object="${user}" th:action="${'/user/edit/'+user.getId()}" method="post"

class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<label >

Введите имя

<input required th:field="\*{login}" type="text" name="login" placeholder="Имя"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"/>

</label>

<div style="color: red" th:if="${#fields.hasErrors('login')}" th:errors="\*{login}"></div>

<br>

<label >

Введите пароль

<input required th:field="\*{password}" type="password" name="password" placeholder="Имя"

class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"/>

</label>

<div style="color: red" th:if="${#fields.hasErrors('password')}" th:errors="\*{password}"></div>

<br>

<br>

<div th:each="role : ${roles}">

<label>

<input th:id="${role}" type="checkbox" th:name="${'roles[]'}" th:value="${role}"

/>

</label>

<label th:text="${role}"/>

<br>

</div>

<div class="flex items-center justify-between">

<button type="submit"

class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline">

Редактировать пользователя

</button>

</div>

</form>

</div>

</body>

</html>

121. resources\templates\user\user-filter.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Filtered users</title>

<meta charset="UTF-8">

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<form class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

<h2 class="font-bold">Фильтрация</h2>

<label>

<input class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 mb-3 leading-tight focus:outline-none focus:shadow-outline"

id="password" type="text" name="title">

</label>

<input class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"

type="submit" formmethod="post" formaction="/user/filter/result" value="Найти">

<br>

<input class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"

type="submit" formmethod="post" formaction="/user/filter\_strict/result" value="Найти (строгий поиск)">

<div th:each="el : ${result}">

<h3 class="font-bold" th:text="${el.login}"></h3>

<p th:text="${el.birthdate}"></p>

</div>

</form>

</div>

</body>

</html>

122. resources\templates\user\user-main.html

<!DOCTYPE HTML>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>User</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

</head>

<body>

<div th:insert="header">...</div>

<div class="w-full max-w-xs">

<h1 class="font-bold">Все пользователи</h1>

<a href="/user/filter" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Поиск

пользователей</a>

<a href="/user/add" class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Добавить

пользователя</a>

<div th:each="el : ${users}">

<div class="inline-block align-baseline rounded w-full py-2 px-3 bg-transparent hover:bg-teal-100 shadow">

<h3 th:text="${el.username}" class="font-bold"></h3>

<p>

<a th:href="'/user/edit/'+${el.getId()}" href="/user/filter"

class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Редактировать</a>

<a th:href="'/user/show/'+${el.getId()}" href="/user/filter"

class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Посмореть</a>

<a th:href="'/user/del/'+${el.getId()}" href="/user/filter"

class="inline-block align-baseline font-bold text-sm text-blue-500 hover:text-blue-800">Удалить</a>

</p>

</div>

<br>

</div>

</div>

</body>

</html>

123. resources\templates\user\user-show.html

<!DOCTYPE html>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

<html xmlns:th="http://www.thymeleaf.org" lang="en">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

<title th:value="${user.login}"></title>

</head>

<body>

<div th:insert="header">...</div>

<h3 th:text="'Логин пользователя '${user.login}"></h3>

</body>

</html>