



НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

CSCB532: Практика по програмиране и Интернет технологии

Изготвила: Ивана Тонева, фн: F111801

Съдържание:

1. Влизане в системата.....	2
1.1Регистрация.....	2
1.2 Вход.....	3
1.3Излизане от системата.....	3
2.Въвеждане, редактиране и изтриване на данни.....	3
2.1 Въвеждане, редактиране и изтриване на данни от таблицата с потребители.	
4	
2.1.1Редактиране.....	4
2.2 Въвеждане, редактиране и изтриване на данни от таблицата с курсове.....	4
2.2.1 Записване.....	4
2.3 Въвеждане, редактиране и изтриване на данни от таблицата със справки...	5
2.3.1 Добавяне.....	5
2.3.2 Редактиране.....	5
2.3.3 Изтриване.....	5

1.Влизане в системата

При отваряне на сайта, потребителят вижда началната страница, която описва за какво е полезен той, както и дава възможност на потребителя да избере дали иска да се регистрира като натисне бутона „Регистрирай се тук“, в случай че няма профил в сайта или да влезе чрез бутона „Вход“, ако вече има такъв.

За различните типове потребители има различна страница, която да им бъде показана, например ако е влезнал преподавател, то той може да вижда информация за себе си и своите студенти, студентът от своя страна може да вижда само своите лични данни, докато администраторът може да вижда цялата налична информация за преподаватели и студенти, както и да определя ролите на регистрираните потребители.

1.1Регистрация

При натискане на бутона „Регистрирай се тук“ се отваря форма за регистрация в сайта съдържаща полета за въвеждане на имена, имейл и парола, както и повтаряне на паролата. След попълване на всички полета с правилни стойности (ако се въведе невалидно потребителско име, парола и/или имейл се изписва съответното съобщение под всяко поле, което информира потребителя какви стойности трябва да бъдат попълнени), се натиска бутона регистрация и потребителят се препраща към „Вход“. Освен това ако потребителя по погрешка е натиснал бутон „Регистрирай се тук“, а той има вече има профил, то може чрез бутона „Вече имаш профил. Цъкни тук!“ да бъде препратен директно към „Вход“.

```
public onRegister(): void {
  if (!!this.registrationForm.controls.userName.errors || {})[!pattern'] || !!this.registrationForm.controls.email.errors || {}[!pattern'] ||
    !!this.registrationForm.controls.password.errors || {}[!pattern'] || !!this.registrationForm.controls.repeatPassword.errors || {}[!pattern']) {
    this.errorMessage = 'Трябва да въведете всички данни за регистрация!';
    return;
  }

  if (this.registrationForm.controls.password.value !== this.registrationForm.controls.repeatPassword.value) {
    this.passwordError = true;
    this.success = false;
  }

  if (this.registrationForm.controls.userName.invalid || this.registrationForm.controls.email.invalid ||
    this.registrationForm.controls.password.invalid || this.registrationForm.controls.repeatPassword.invalid) {
    return;
  }

  const user: User = new User();
  user.id = uuid.v4();
  user.username = this.registrationForm.controls.userName.value || '';
  user.password = this.registrationForm.controls.password.value || '';
  user.email = this.registrationForm.controls.email.value || '';
  user.role = this.registrationForm.controls.role.value || this.USER_ROLES.STUDENT.toString();
  user.firstName = this.registrationForm.controls.firstName.value || '';
  user.lastName = this.registrationForm.controls.lastName.value || '';
  this.userRepository.registerUser(user).subscribe(() => {
    this.router.navigateByUrl('login-page');
    this.success = true;
    this.errorMessage = '';
  }, (err) => {
    this.errorMessage = err.error.message;
  });
}

public haveAProfile(): void {
  this.router.navigateByUrl('login-page');
}
```

1.2 Вход

За да бъде използвана системата от даден потребител, то той първо трябва да „влезе“ в нея, за целта при натискане на бутона „Вход“ от началната страница, се отваря форма, която съдържа полета за въвеждане на потребителско име и парола. При въвеждане на правилни данни (ако се въведе невалидно потребителско име и/или парола се изписва съответното съобщение, което информира потребителя за грешка), потребителят успешно влиза в системата.

Ако потребителят няма регистрация, то при натискане на бутона „Нямаш профил? Регистрирай се тук!“ ще бъде препратен към регистрационната форма, за да си създаде профил.

Ако няма регистриран потребител с тези данни се показва подходящо съобщение за грешка.

```
public onLogin(): void {  
  if (!!this.loginForm.controls.userName.errors || {})[!pattern'] || (!!this.loginForm.controls.password.errors || {})[!pattern']) {  
    this.errorMessage = 'Липсват данни за вход!';  
    return;  
  }  
  
  if (this.loginForm.controls.userName.touched) {  
    this.errorMessage = '';  
  }  
  
  if (!this.loginForm.controls.userName.valid || !this.loginForm.controls.password.valid) {  
    return;  
  }  
  
  const user: User = new User();  
  user.username = this.loginForm.controls.userName.value || '';  
  user.password = this.loginForm.controls.password.value || '';  
  
  this.userRepository.loginUser(user).subscribe((response: User) => {  
    this.userService.loginUser(response);  
    this.router.navigateByUrl('main-page');  
    this.isLoginButtonVisible = false;  
    this.isLogoutButtonVisible = true;  
  }, (err) => {  
    this.errorMessage = err.error.message;  
  })  
}  
  
public onRegistration(): void {  
  this.router.navigateByUrl('register-page');  
}
```

1.3 Излизане от системата

За да излезе от системата, потребителят трябва да натисне бутона „Изход“, който го отвежда отново на началната страница на системата.

```
onLogoutButton(): void {  
  this.userService.logoutUser();  
  this.router.navigate(['']);  
}
```

2. Въвеждане, редактиране и изтриване на данни

В зависимост от ролята на потребителя се показват различни менюта, които той да вижда. Например потребител с роля „Администратор“, има достъп до всички възможни менюта без „Курсове“, но най-важно е менюто съдържащо информация за всички регистрирани потребители. Потребител с роля „Преподавател“ няма достъп до менюто с потребители, но има до менюта като „Справки“ и „Тестова система“. Потребител с роля „Студент“ има достъп до менюто „Курсове“ и „Справки“.

2.1 Въвеждане, редактиране и изтриване на данни от таблицата с потребители

Тази таблица е видима само за потребителите с роля „Администратор“ и съдържа полета с цялата информация за даден потребител при неговата регистрация. Администраторът има право да редактира потребител, като е неговата задача да определи кой потребител каква роля има в системата. Всеки новорегистриран потребител влиза в система с роля „Студент“.

2.1.1 Редактиране

Чрез натискането на бутона „Редактиране“ се показва форма с полета, с попълнените данни на съответния потребител като администраторът има право да смени всичко освен потребителското му име. Паролата не е включена към данните, които се показват.

```
onEditClick(user: User): void {
    this.selectedUser = user;
    this.registrationForm.controls.firstName.setValue(user.firstName || '');
    this.registrationForm.controls.lastName.setValue(user.lastName || '');
    this.registrationForm.controls.userName.setValue(user.username || '');
    this.registrationForm.controls.password.setValue(user.password || '');
    this.registrationForm.controls.email.setValue(user.email || '');
    this.registrationForm.controls.role.setValue(user.role || UserRoles.STUDENT.toString());
    this.showEditUserForm = true;
    this.showUserPage = false;
}

onEditUser(): void {
    this.selectedUser.username = this.registrationForm.controls.userName.value || '';
    this.selectedUser.password = this.registrationForm.controls.password.value || '';
    this.selectedUser.email = this.registrationForm.controls.email.value || '';
    this.selectedUser.role = this.registrationForm.controls.role.value || UserRoles.STUDENT.toString();
    this.selectedUser.firstName = this.registrationForm.controls.firstName.value || '';
    this.selectedUser.lastName = this.registrationForm.controls.lastName.value || '';

    this.userRepository.editUser(this.selectedUser).subscribe((response) => {
        this.userData.forEach(user => {
            if (user.id === response.id) {
                user = response;
            }
        })
    });
    this.showEditUserForm = false;
    this.showUserPage = true;
};
```

2.2 Въвеждане, редактиране и изтриване на данни от таблицата с курсове

Тази таблица е видима само за потребителите с роля „Студент“ като те я използват, за да се запишат за съответните курсове, които искат. Таблицата съдържа полета с цялата информация за даден курс като име, материали, задача и съответния преподавател.

2.2.1 Записване

При натискане на бутона „Запиши“ пред потребителя се отваря модал, който го информира, че курсът който е избрал ще бъде записан дали той е съгласен с това. При натискане на бутона „Отказ“, потребителят се отказва от записването на курса, а при натискане на бутона „Потвърждаване“, курсът се записва в менюто „Справки“ и се скрива от менюто с наличните курсове.

```
ngOnInit() {
  this.userRepository.fetchAllCoursesForStudentToEnroll().subscribe((response) => {
    this.coursesData = response;
  })
}

onAddClick(course: Subject): void {
  this.selectedCourse = course;
  this.showAddSubjectForm = true;
}

onCancelClick(): void {
  this.showAddSubjectForm = false;
  this.showCoursePage = true;
}

onSubmitCourse(): void {
  this.userRepository.enrollStudentForSubject(this.selectedCourse)
    .pipe(finalize(() => {
      this.showAddSubjectForm = false;
      this.showCoursePage = true;
    }))
    .subscribe(() => {
      this.coursesData = this.coursesData.filter(s => s.id !== this.selectedCourse.id);
      this.showSuccessfulAddSubjectForm = true;
      this.showCoursePage = false;
    })
}

onOkClick() {
  this.showSuccessfulAddSubjectForm = false;
  this.showCoursePage = true;
}
```

2.3 Въвеждане, редактиране и изтриване на данни от таблицата със справки

Тази таблица е видима за всички потребителите, но те имат различен достъп до нея. Потребителят с роля „Администратор“ може да добавя, редактира и изтрива всички курсове, потребителят с роля „Преподавател“ може да добавя, редактира и изтрива само собствените си курсове, докато потребител с роля „Студент“ може само да види записаните от него курсове. Таблицата съдържа полета с цялата информация за даден курс, като администраторът и преподавателя може да виждат и записаните студенти към съответните курсове.

2.3.1 Добавяне

При натискане на бутона „Добави“ пред потребителите с роля „Администратор“ или „Преподавател“ се отваря форма за попълване на данни за нов курс, който да бъде добавен към системата.

```
onAddClick(): void {
  this.showAddSubjectForm = true;
  this.showUserPage = false;
  this.showEditUserForm = false;
}

onAddCourse(): void {
  const newSubject: Subject = new Subject();
  newSubject.name = this.subjectForm.controls.name.value || '';
  newSubject.materials = this.subjectForm.controls.materials.value || '';
  newSubject.tasks = this.subjectForm.controls.tasks.value || '';
  if (!newSubject.name || !newSubject.materials || !newSubject.tasks) {
    this.shouldShowEmptyFieldsError = true;
    return;
  }

  newSubject.teacher = {username: this.userService.getCurrentUser()} as User
  this.userRepository.addCourse(newSubject).subscribe((response) => {
    this.coursesData.push(response);
  });

  this.showAddSubjectForm = false;
  this.showEditUserForm = false;
  this.showUserPage = true;
}
```

2.3.2 Редактиране

При натискане на бутона „Редактиране“ пред потребителите с роля „Администратор“ или „Преподавател“ се отваря форма с попълнените данни за даден курс, като те могат да променят цялата информация за даден курс, освен да сменят името на преподавателя.

```
onEditClick(course: Subject): void {
  this.selectedCourse = course;
  this.subjectForm.controls.name.setValue(course.name || '');
  this.subjectForm.controls.teacher.setValue((course.teacher.firstName || '') + " " + (course.teacher.lastName || ''));
  this.subjectForm.controls.students.setValue(this.getStudentsLabel(course));
  this.subjectForm.controls.materials.setValue(course.materials || '');
  this.subjectForm.controls.tasks.setValue(course.tasks || '');

  this.showEditUserForm = true;
  this.showUserPage = false;
  this.showAddSubjectForm = false;
}

onEditCourse(): void {
  this.selectedCourse.name = this.subjectForm.controls.name.value || '';
  this.selectedCourse.materials = this.subjectForm.controls.materials.value || '';
  this.selectedCourse.tasks = this.subjectForm.controls.tasks.value || '';
  this.selectedCourse.teacher = this.subjectForm.controls.teacher.value || '';
  this.selectedCourse.students = this.subjectForm.controls.students.value || '';

  this.userRepository.editCourse(this.selectedCourse).subscribe((response) => {
    this.coursesData.forEach(course => {
      if (course.id === response.id) {
        course = response;
      }
    })
  });
  this.showEditUserForm = false;
  this.showUserPage = true;
  this.showAddSubjectForm = false;
};

onCancelClick(): void {
  this.showEditUserForm = false;
  this.showAddSubjectForm = false;
  this.showUserPage = true;
}
```

2.3.3 Изтриване

Чрез натискането на бутона „Изтриване“ потребителят с роля „Администратор“ или „Преподавател“ може да изтрие вече съществуващ курс от таблицата.

```
onDeleteCourse(course: Subject): void {  
  this.userRepository.deleteCourse(course).subscribe(() => {  
    this.coursesData = this.coursesData.filter(c => c.id !== course.id)  
  });  
}
```