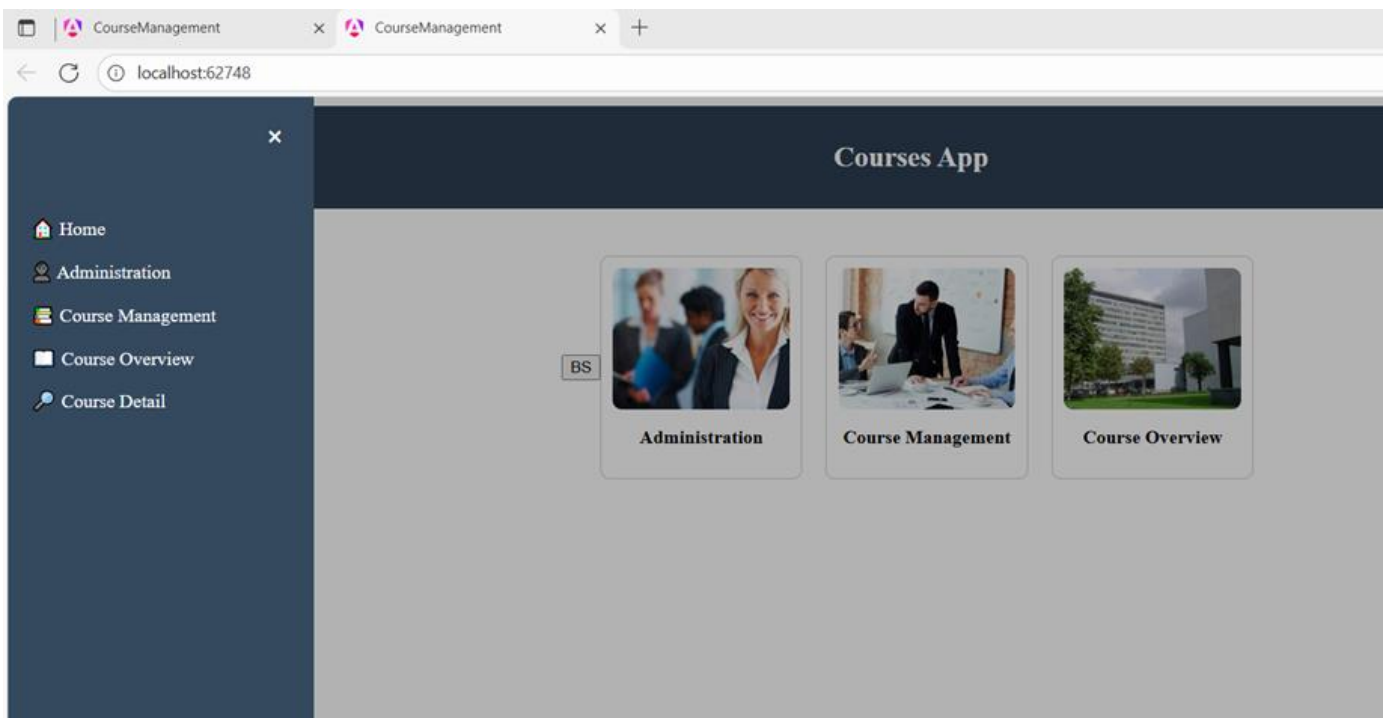


Project Courses App

Courses App is a web application for managing courses, users, and departments. The frontend is implemented in Angular using TypeScript, providing a responsive and dynamic user interface with routing and bilingual support (English/Bosnian). The backend is built with Java Spring Boot, exposing REST APIs for CRUD operations. Data is stored in a MySQL database managed locally via XAMPP and MySQL Workbench, which allows database design, query execution, and data management. HikariCP is used for efficient connection pooling between the application and the database.

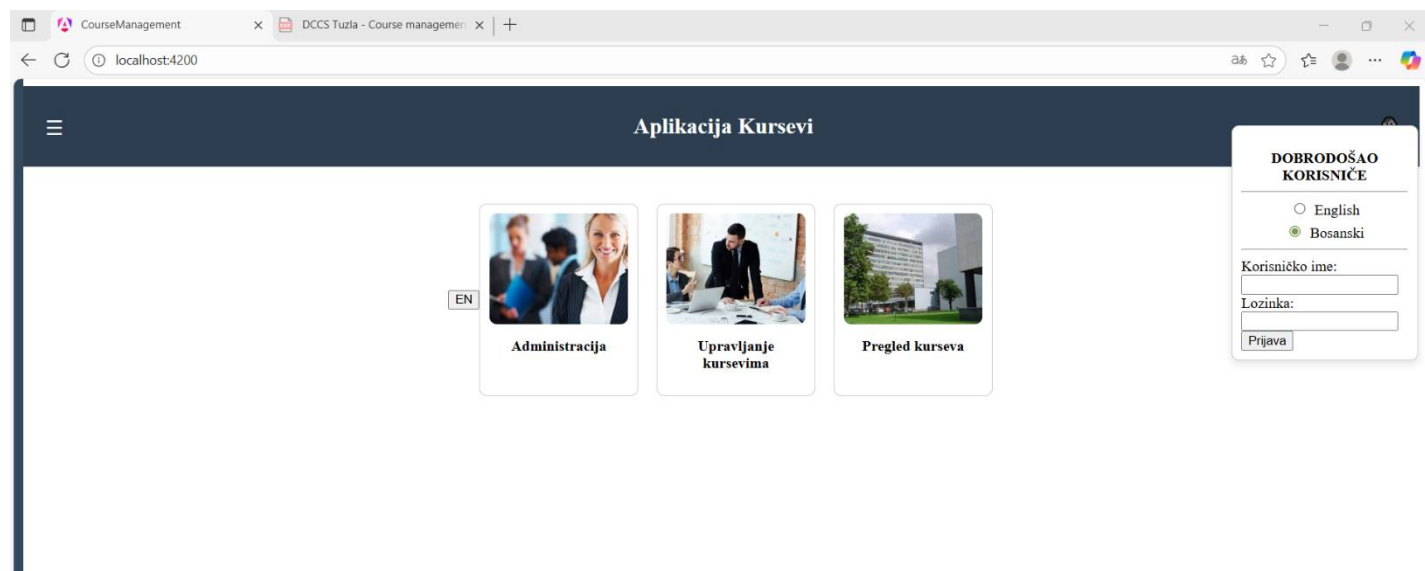
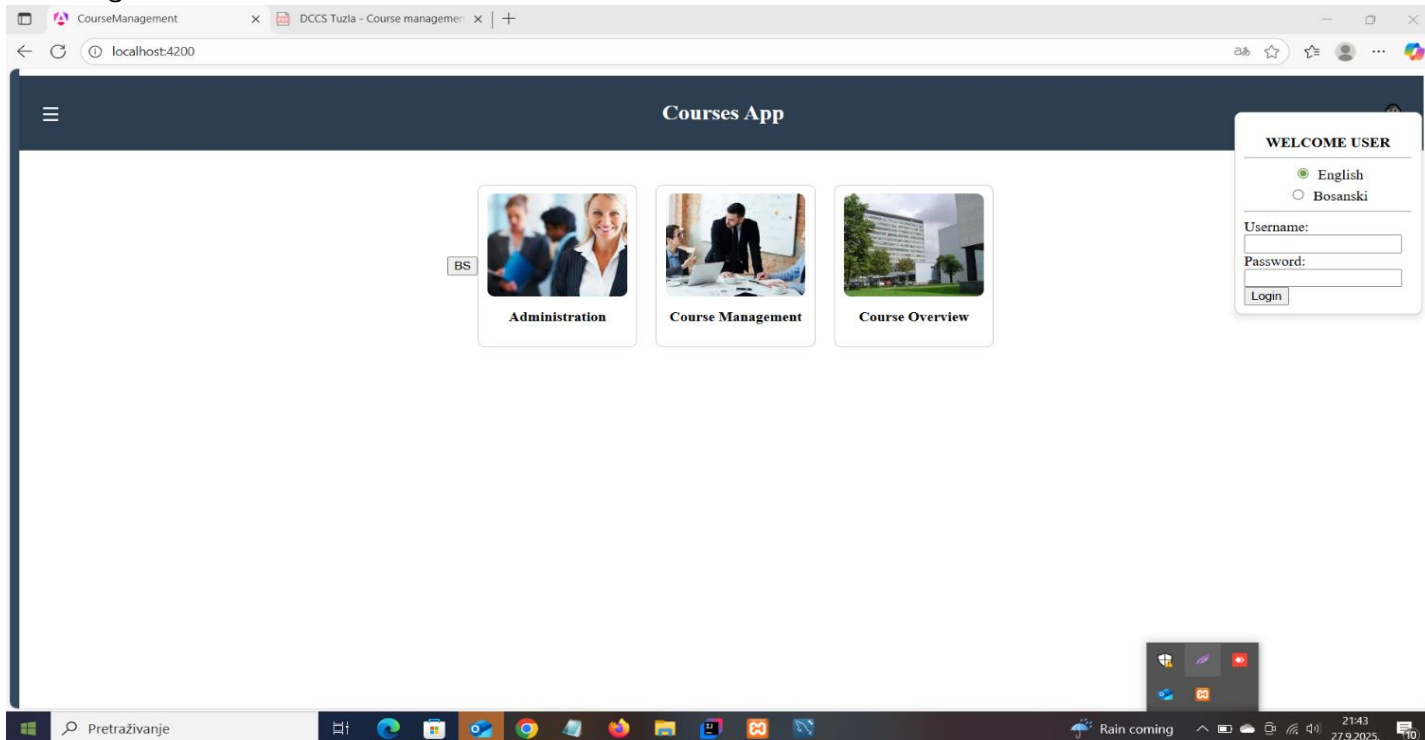
AppComponent

The main layout component integrates the header, sidebar navigation, user menu, and content area (router-outlet). It manages language toggling, simulated authentication, and provides a central layout for all other components. It ensures consistent navigation and a responsive interface throughout the application.



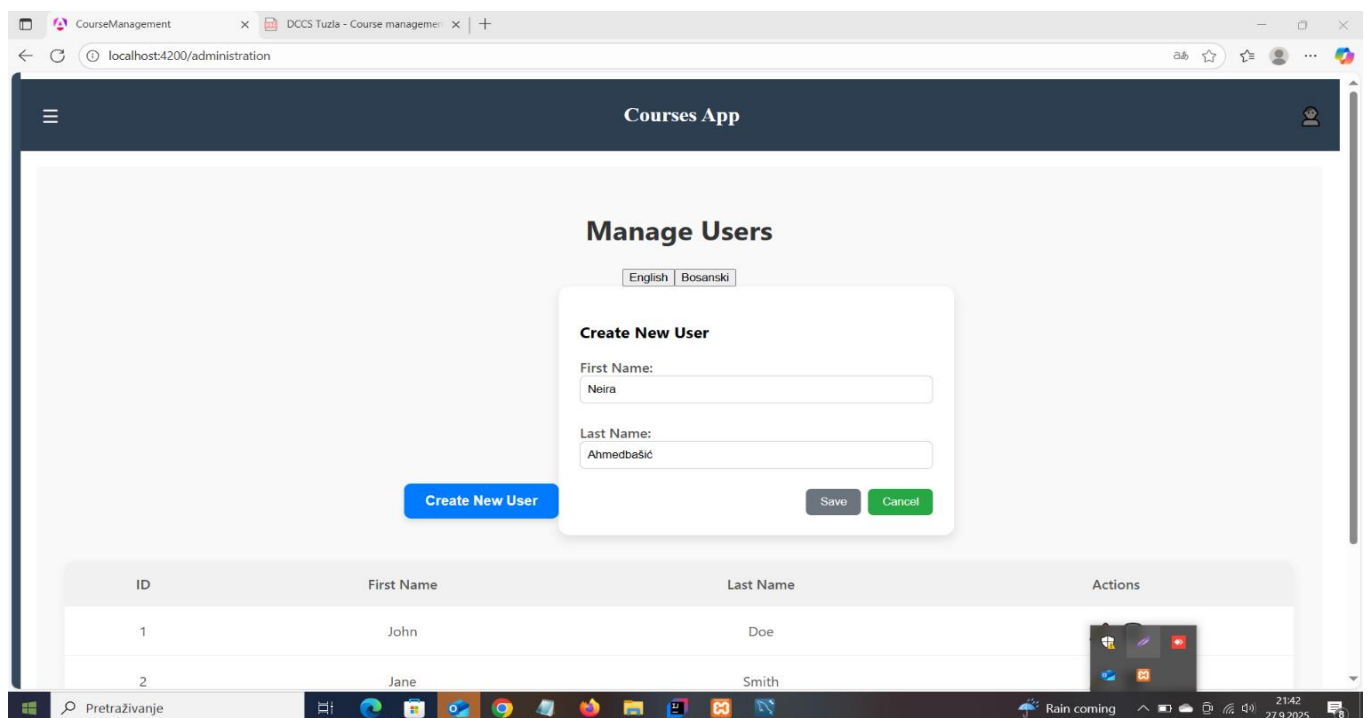
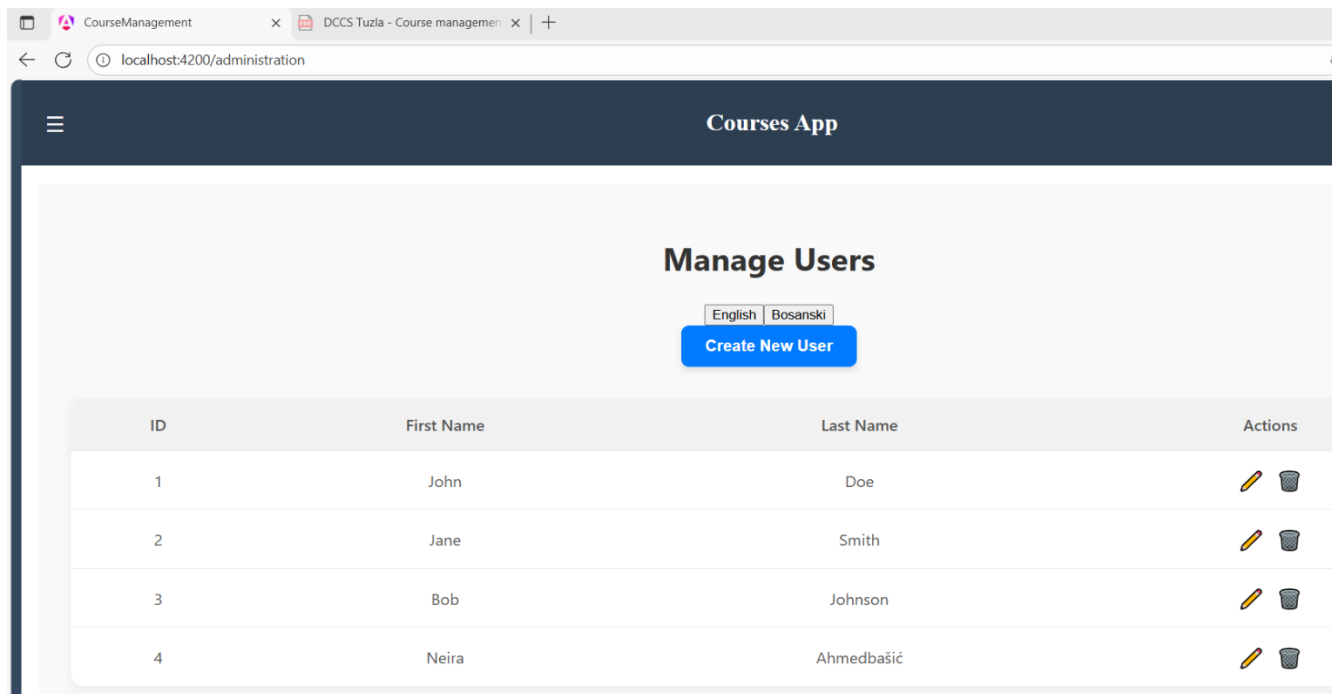
HomeViewComponent

This is the landing page, providing quick access to the main modules: Administration, Course Management, and Course Overview. Users can toggle the language between English and Bosnian. The interface uses card-style modules for a clear visual overview and intuitive navigation.



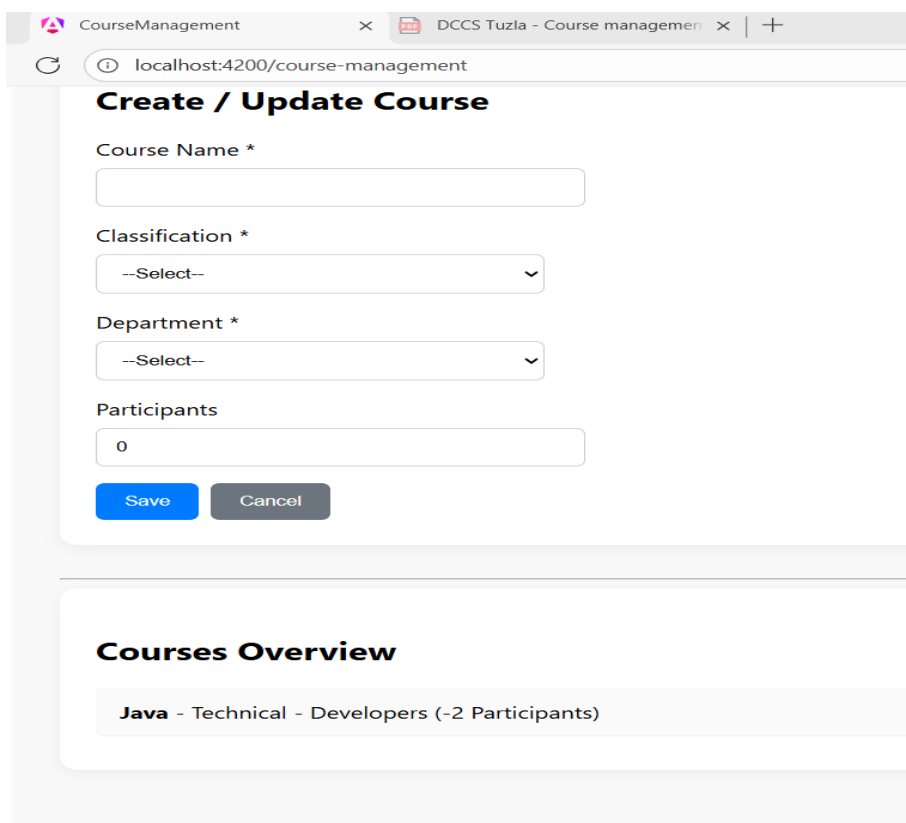
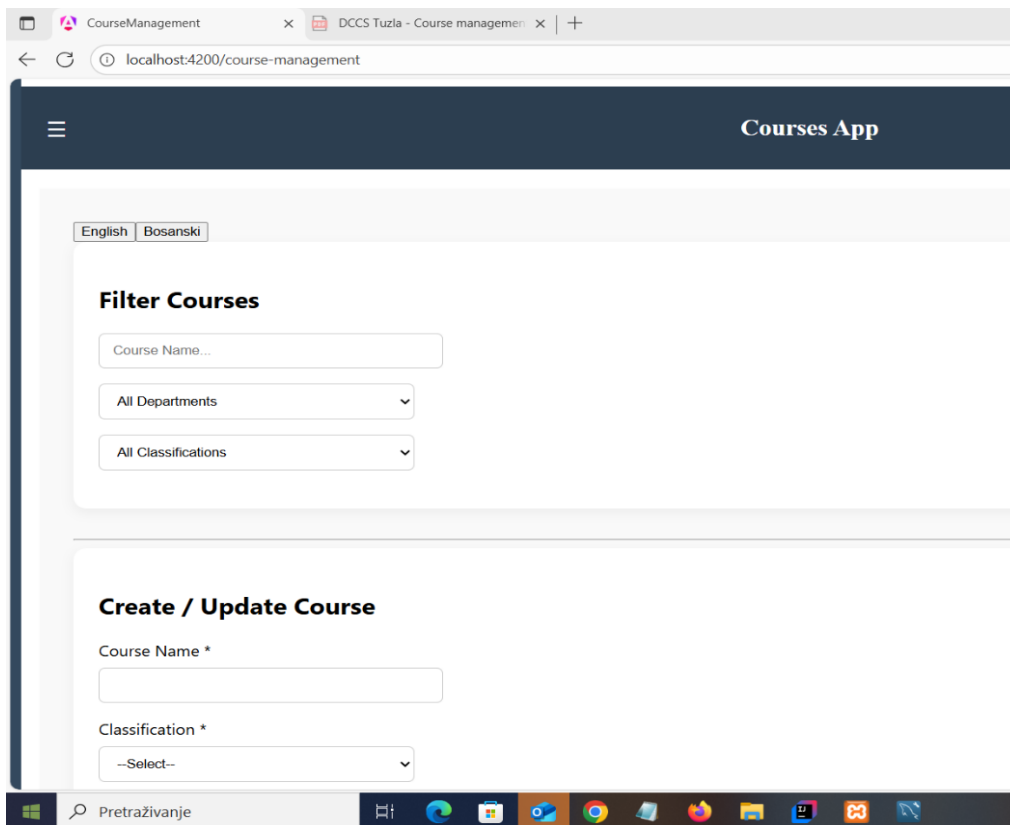
AdministrationComponent

This component handles user management. It allows viewing, creating, editing, and deleting users. Dialog forms with validation are used for user creation and editing. Language switching is supported dynamically. Users are managed via mock data, with automatic ID assignment, and the layout is responsive and clear.



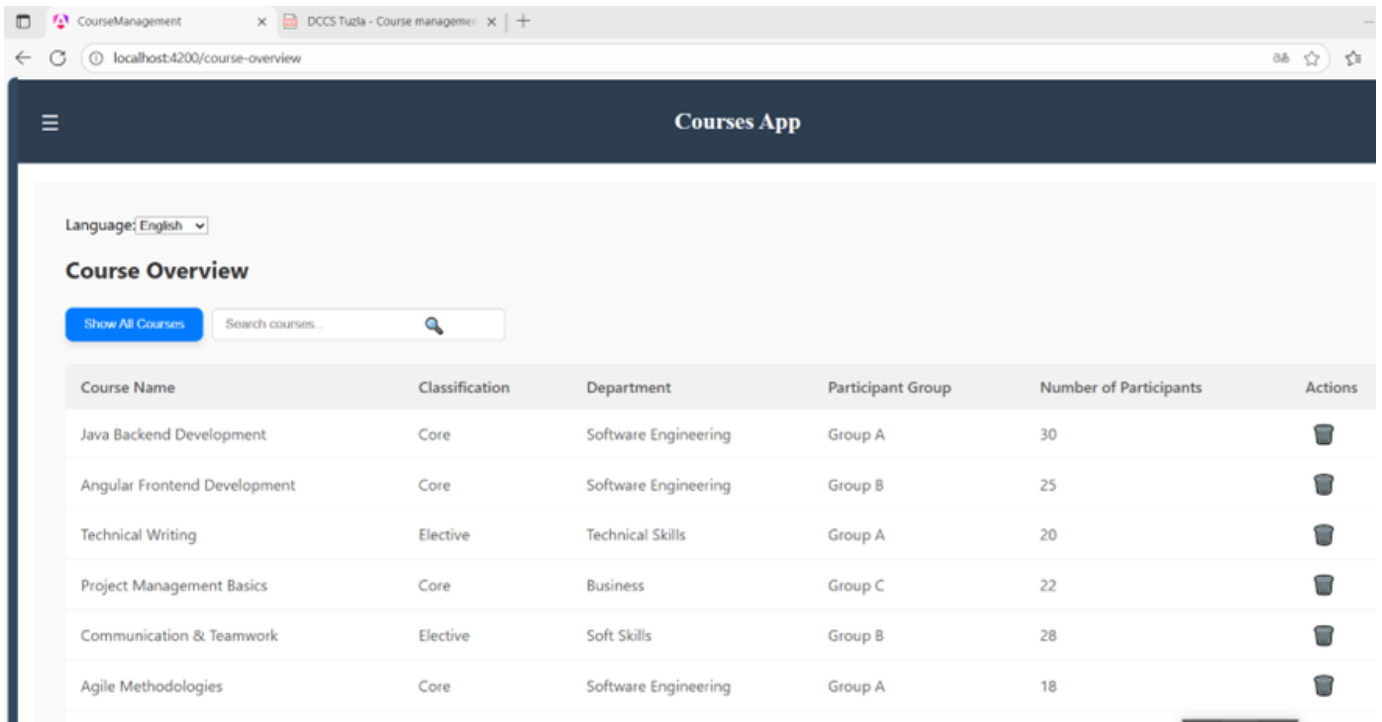
CourseManagementComponent

CourseManagementComponent enables creating, updating, and filtering courses. Users can search courses by name, department, or classification, create or update course details, and view a dynamic list of all courses with essential information. The interface is interactive, bilingual, and includes form validation for required fields.









CourseOverviewComponent

This component provides an overview of all courses, with details including name, classification, department, participant group, and number of participants. Users can search and filter courses dynamically, delete courses, and switch between English and Bosnian. Data is managed locally within the component, with translations included for bilingual support.

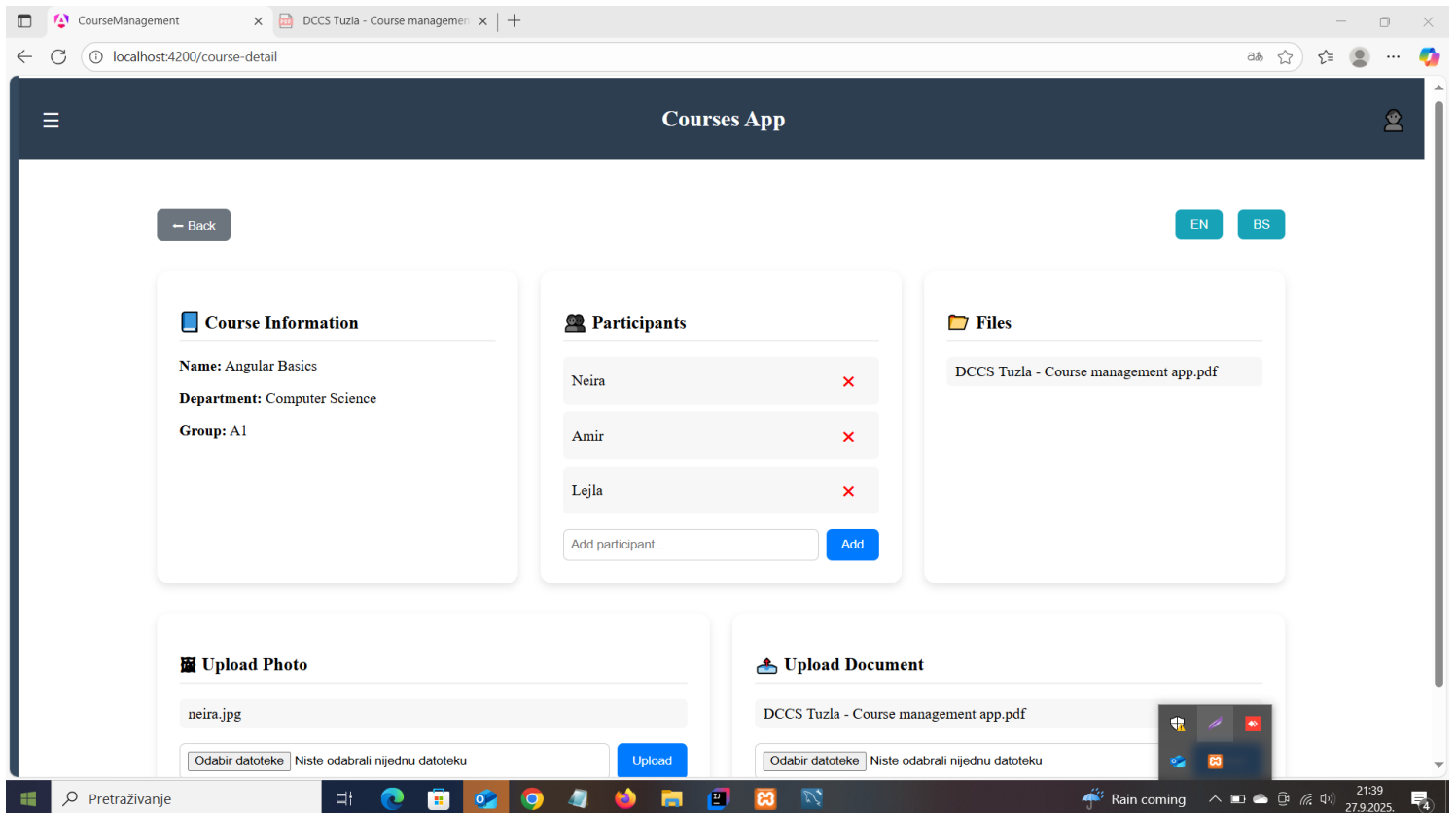


The screenshot displays a web application titled "Courses App" running on a browser at localhost:4200/course-overview. The interface includes a language selector set to "English" and a "Course Overview" section with a "Show All Courses" button and a search bar. Below this is a table listing six courses with columns for Course Name, Classification, Department, Participant Group, Number of Participants, and Actions. Each course has a trash icon in the Actions column for deletion.

Course Name	Classification	Department	Participant Group	Number of Participants	Actions
Java Backend Development	Core	Software Engineering	Group A	30	
Angular Frontend Development	Core	Software Engineering	Group B	25	
Technical Writing	Elective	Technical Skills	Group A	20	
Project Management Basics	Core	Business	Group C	22	
Communication & Teamwork	Elective	Soft Skills	Group B	28	
Agile Methodologies	Core	Software Engineering	Group A	18	

CourseDetailComponent

CourseDetailComponent displays detailed information about a specific course, including course info, participants, documents, and photos. Users can add or remove participants, upload documents and photos, and navigate back to the course overview. The layout uses sectioned cards for clarity, with dynamic lists managed via Angular directives, and supports bilingual labels and content.



Database Integration

The backend connects to a MySQL database hosted locally through XAMPP. Database tables include User, Role, Department, Course, CourseClassification, and CourseParticipant. All data operations are performed via Spring Boot repositories. MySQL Workbench is used to design, query, and manage tables, ensuring data integrity and efficient management. The REST API exposes the data to the Angular frontend.

The screenshot displays three database tables in MySQL Workbench:

- User Table:** Contains user information including id, first_name, last_name, username, password, and role.
- Department Table:** Contains department information including id and name.
- Course Table:** Contains course information including id, title, description, duration_hours, department_id, classification_id, and participants.

id	first_name	last_name	username	password	role
1	John	Doe	admin	admin	ADMIN
2	Jane	Smith	janesmith	password123	USER
3	Bob	Johnson	bobjohnson	password123	USER
4	Neira	Ahmedbašić	neira123	lozinka123	USER
5	Amir	Kovačević	amirk	pass123	USER
6	Elma	Hadžić	elmah	pass123	USER
NULL	NULL	NULL	NULL	NULL	NULL

id	name
1	IT
2	HR
3	Finance
4	Marketing
5	Sales
NULL	NULL

id	name
1	Beginner
2	Intermediate
3	Advanced
4	Expert
NULL	NULL

id	title	description	duration_hours	department_id	classification_id	participants
1	Angular Basics	Learn the basics of Angular	20	1	1	10
2	Java Basics	Introductory Java course	40	1	1	15
3	Spring Boot Fundamentals	Building REST APIs with Spring Boot	30	1	2	12
4	Database Design	Learn relational database concepts and SQL	25	1	2	18
5	Web Development with Angular	Modern frontend development using Angular	35	1	2	10
6	Cloud Computing	Introduction to cloud services and architecture	30	1	3	9
7	Effective Communication	Improve workplace communication skills	15	2	1	20
8	Teamwork & Collaboration	How to work effectively in teams	12	2	1	18
9	Leadership Essentials	Develop basic leadership and management skills	20	2	2	14
10	Time Management	Organize and prioritize tasks effectively	10	2	1	25
11	Conflict Resolution	Learn strategies to resolve workplace conflicts	15	2	2	11
12	Project Management	Basics of project management	15	2	2	22

Frontend Functionality

The Angular frontend provides dynamic course management, participant handling, file uploads, and responsive design. All UI elements are bilingual, and the interface adapts seamlessly to desktop and mobile screens. Components interact with the backend via REST APIs to fetch and update data.

[Postman-Courses App API](#)