



IT 309 SOFTWARE ENGINEERING

PROJECT DOCUMENTATION

radnikba

Prepared by:

Harun Kunovac

Ervin Vladić

Proposed to:

Nermina Durmić, Assist. Prof. Dr.
Aldin Kovačević, Teaching Assistant

Date of submission

21.06.2023

Table of Contents

- Introduction3**
 - About the project3
 - Project Functionalities and Screenshots3
- Project Structure7**
 - Technologies7
 - Database Entities7
 - Design Patterns.....7
 - Tests.....8
- Conclusion8**


1. Introduction

1.1. About the Project

Radnikba is an online platform for advertising construction workers. Aware of the lack of construction workers in our country, we believe that a platform is needed so that insight into all workers is more easily accessible. Users of the platform will be able to see all the basic information about workers.

1.2. Project Functionalities and Screenshots

- Functional login
- Functional register
- Landing page
- Homepage where jobs can be seen
- Display for all workers who advertise on the network and basic information (Name, surname, contact phone)



Dobrodošli na Radnikba

Email adresa

[Registruj se...](#)

Login page

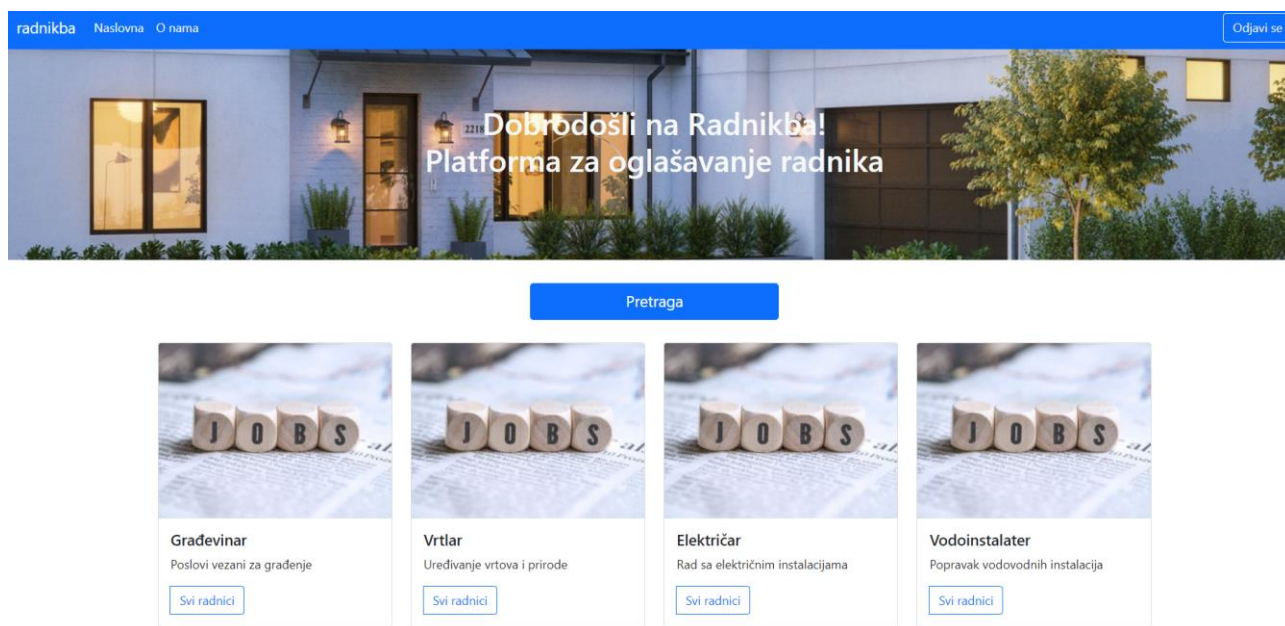


Registruj se na Radnikba

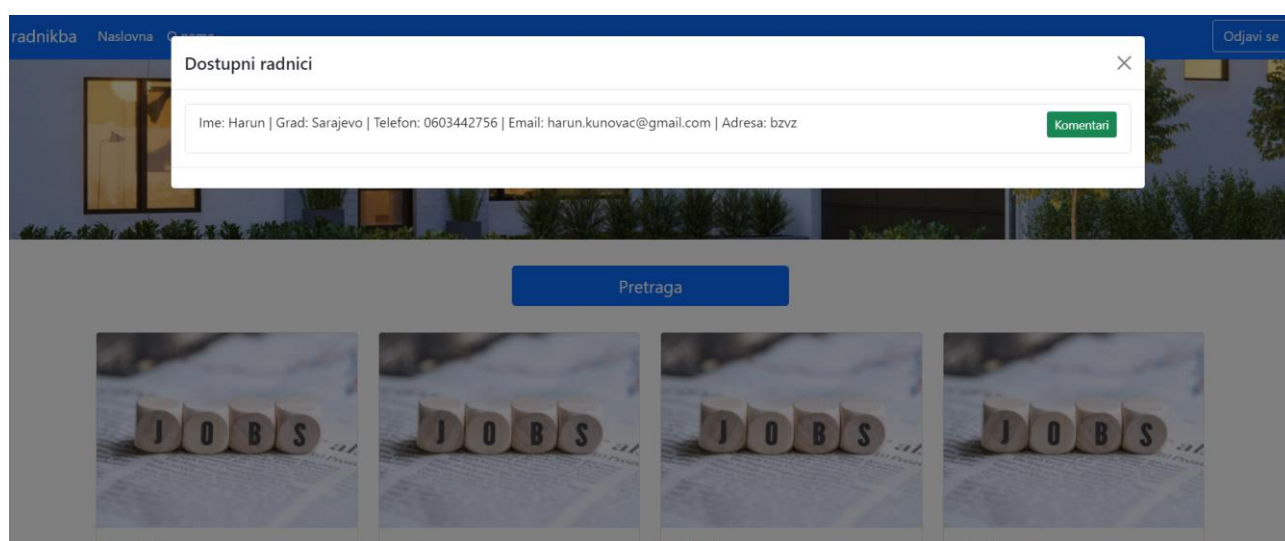
Registruj se

[Nazad na Login](#)

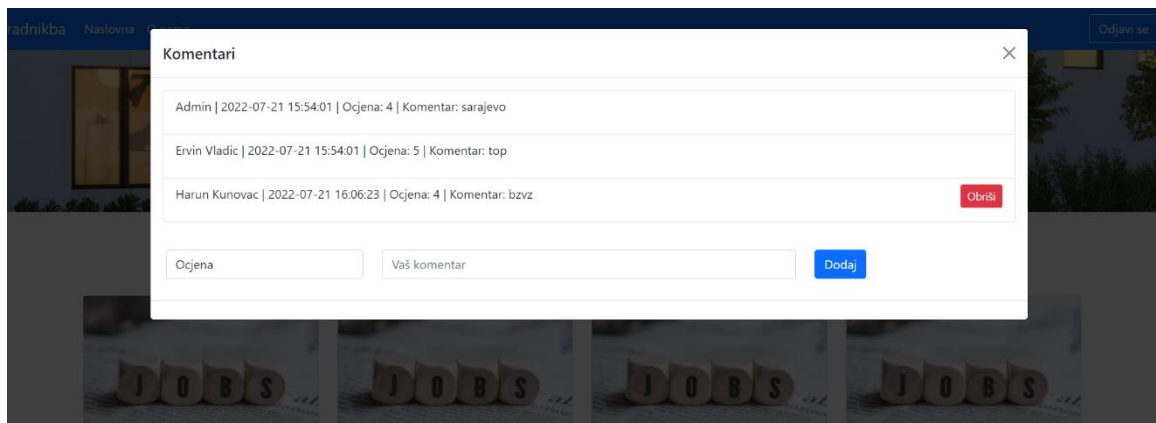
Registration page



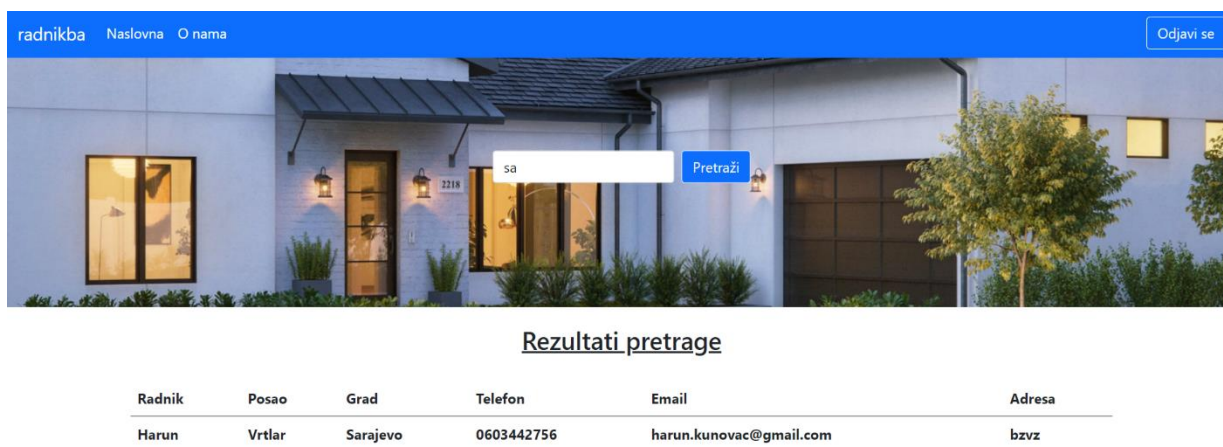
Main page as a user



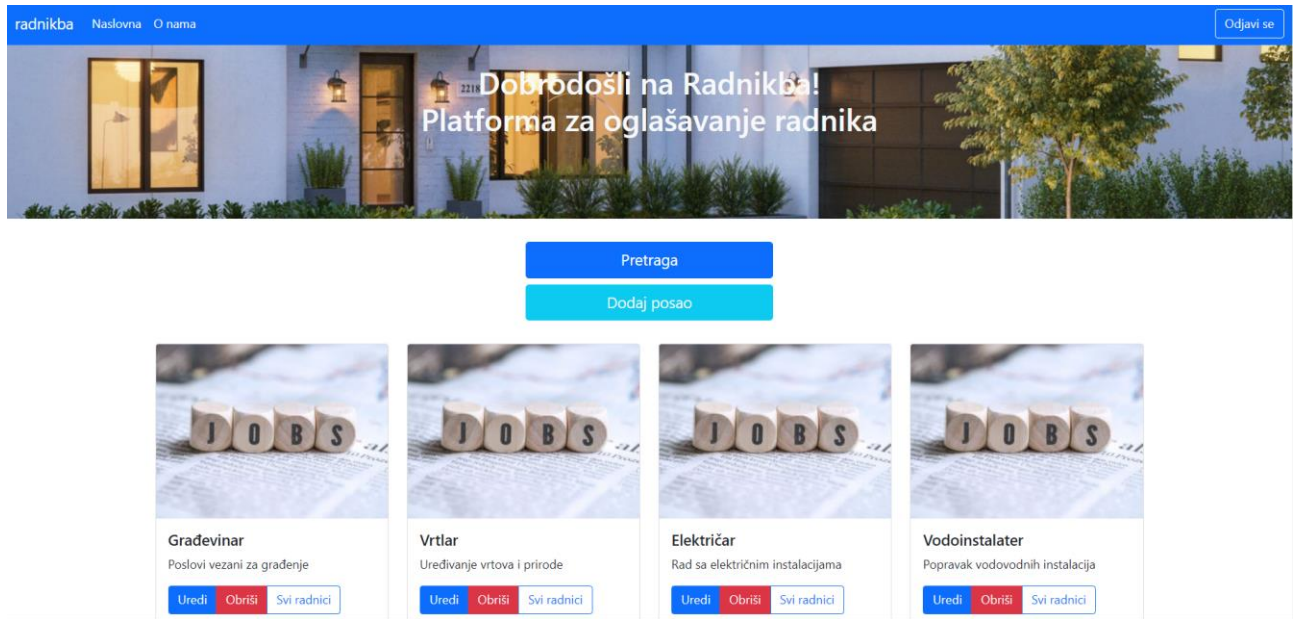
Insight into workers



View, add, and delete comments



Search option



Main page as admin

2. Project Structure

2.1. Technologies

For technologies we used: HTML, CSS, Bootstrap 5, JavaScript, PHP, jQuery, FlightPHP, SQL, Selenium for testing, and for coding standards we used: PSR-1: Basic Coding Standard, PSR-12: Extended Coding Style

2.2. Database Entities

- Users - All users with user information such as user name, user surname, city, email, and password
- Job - All jobs, including the job name and job description
- Worker - All workers, including information such as worker name, worker city, worker phone number, worker address, and worker email
- Review - All reviews given by the user to the employee, including the review grades and review comments

2.3. Design Patterns

- MVC (Model-View-Controller) Pattern

We used MVC pattern across our project to help separate concerns and promote code reusability.

Model: The model represents the data layer of our application and it consists of DAO (Data Access Object) classes, which handle the interaction with the database. These classes are located inside *rest/dao* folder.

View: The view represents the user interface (HTML, CSS, and JavaScript) that the users interact with. In our application, *index.html*, *search.html* and *login.html* represent views.

Controller: The controller acts as an intermediary between the model and the view. It handles user input, processes requests, and updates the model and view accordingly. In our code we used FlightPHP to setup endpoints to receive requests, and interact with the appropriate service located in *services* folder, which then sends the request to appropriate DAO class.

- **Data Access Object (DAO) design pattern**

The DAO design pattern separates the data persistence logic from the business logic of the application. It provides an abstraction layer between the application code and the database, allowing us to perform CRUD operations (Create, Read, Update, Delete) on all entities without directly interacting with the database.

In our case, we have a base DAO class that contains the general CRUD operations implemented. This class serves as the foundation for entity-specific DAO classes, which extend the base class and can add new functions if needed. This hierarchical structure allows us to reuse the common data manipulation operations across different entities while providing flexibility to add entity-specific functionality. This structure leads to easier maintenance and scalability of the application's data layer. (*rest/dao*, *rest/services*)

2.4. Tests

For this project we wrote Selenium tests. The tests we have done are:

-loginTest() - User login test

-registerTest() - User registration test

-reviewTest() - Test if the user has visibility into workers and the ability to add and delete reviews

-searchTest() - Testing the search option

-adminTest() - Admin login and testing of adding, editing and deleting jobs

-adminTest2() - Test for adding and deleting workers as an administrator

The tests are located in the selenium-testing directory. The directory also contains detailed testing documentation.

3. Conclusion

While doing this project, we went through different phases of app development including planning, designing, implementation, deployment and testing at the end. We think that we have successfully mastered the material and that we have fulfilled the initial requirements we set on the beginning. Of course, with more time, we would have a greater focus on details. The next step in the application development would for sure be adding new functionalities to make it more usable. All things

considered, we are grateful for the opportunity to show our knowledge that we have gained in previous years on a project of this type.