



**Grindin'**

# Table of Contents

|   |    |
|---|----|
| I. About me .....                               | 3  |
| II. The idea of the project .....               | 3  |
| III. Stages of work.....                        | 5  |
| Backend Foundations .....                       | 5  |
| Angular Setup & Authentication .....            | 5  |
| User-Side Functionality.....                    | 5  |
| Admin Features & Preparation.....               | 5  |
| IV. Used Technologies.....                      | 6  |
| V. Block Scheme .....                           | 7  |
| VI. Entity-Relationship Diagram .....           | 8  |
| VII. Installation (to run the source code)..... | 9  |
| 1. Prerequisites.....                           | 9  |
| 2. Clone the Repository.....                    | 9  |
| 3. Backend Setup (ASP.NET Core) .....           | 9  |
| 4. Frontend Setup (Angular).....                | 10 |
| 5. Admin Setup .....                            | 11 |
| 6. Notes.....                                   | 11 |
| VIII. Pictures of UI .....                      | 12 |
| • Login Form.....                               | 12 |
| • Registration Form.....                        | 12 |
| • Home Page .....                               | 13 |
| • Profile Page .....                            | 13 |
| • Job Ads Page .....                            | 14 |
| • Job Applications Page.....                    | 14 |

## I. About me

| Name                     | Role                                     | Grade / Degree             |
|--------------------------|--|----------------------------|
| <b>Bozhidar A. Dimov</b> | Project Creator and Full-Stack Developer | 11 V / Applied programming |

## II. The idea of the project

**Grindin'** is a modern and responsive web application designed to streamline the process of tracking, managing, and reviewing job applications. Developed using Angular and TypeScript on the frontend and powered by .NET on the backend, the platform provides a user-friendly interface for job seekers and administrators alike. Whether you're applying to multiple companies or managing submitted applications, **Grindin'** simplifies the job search experience and helps users stay organized and efficient.

The application allows users to:

- Register and manage personal profiles
- Browse and view detailed job advertisements
- Submit applications directly through the platform
- Track application statuses
- Access personalized dashboards

- Create and manage job postings (admin functionality)
- Review and evaluate submitted applications (admin)

With role-based access control, robust form validation, and modern UI components, **Grindin'** offers a secure, accessible, and intuitive experience for both regular users and administrators. The clean separation of modules and the use of Angular best practices ensure maintainability and scalability for future development.

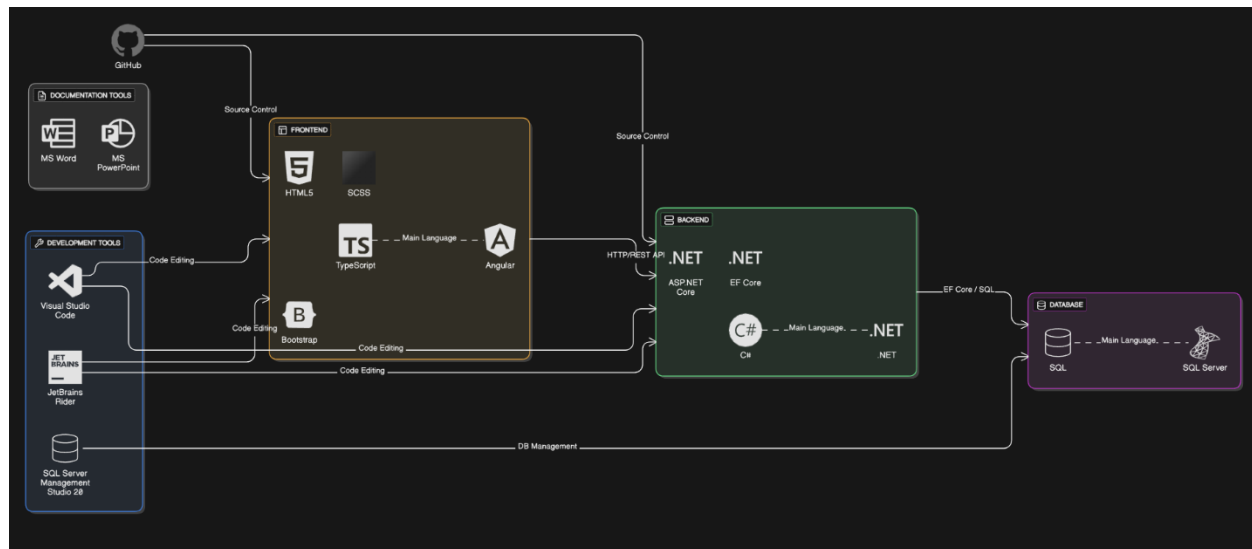
This project was developed as part of the Production Practice curriculum (2024–2025) in class 11 V at the Vocational School of Computer Programming and Innovation (VSCPI) in Burgas, Bulgaria. It serves as both a practical demonstration of software engineering skills and a full-stack web development project, showcasing technologies such as Angular, ASP.NET, Bootstrap, and RESTful API integration.

**Grindin'** reflects the application of theoretical knowledge into a real-world scenario, emphasizing good coding standards, user-centric design, and maintainable architecture. It was completed under the guidance of academic mentors as a milestone in the professional development of the students involved.

### III. Stages of work

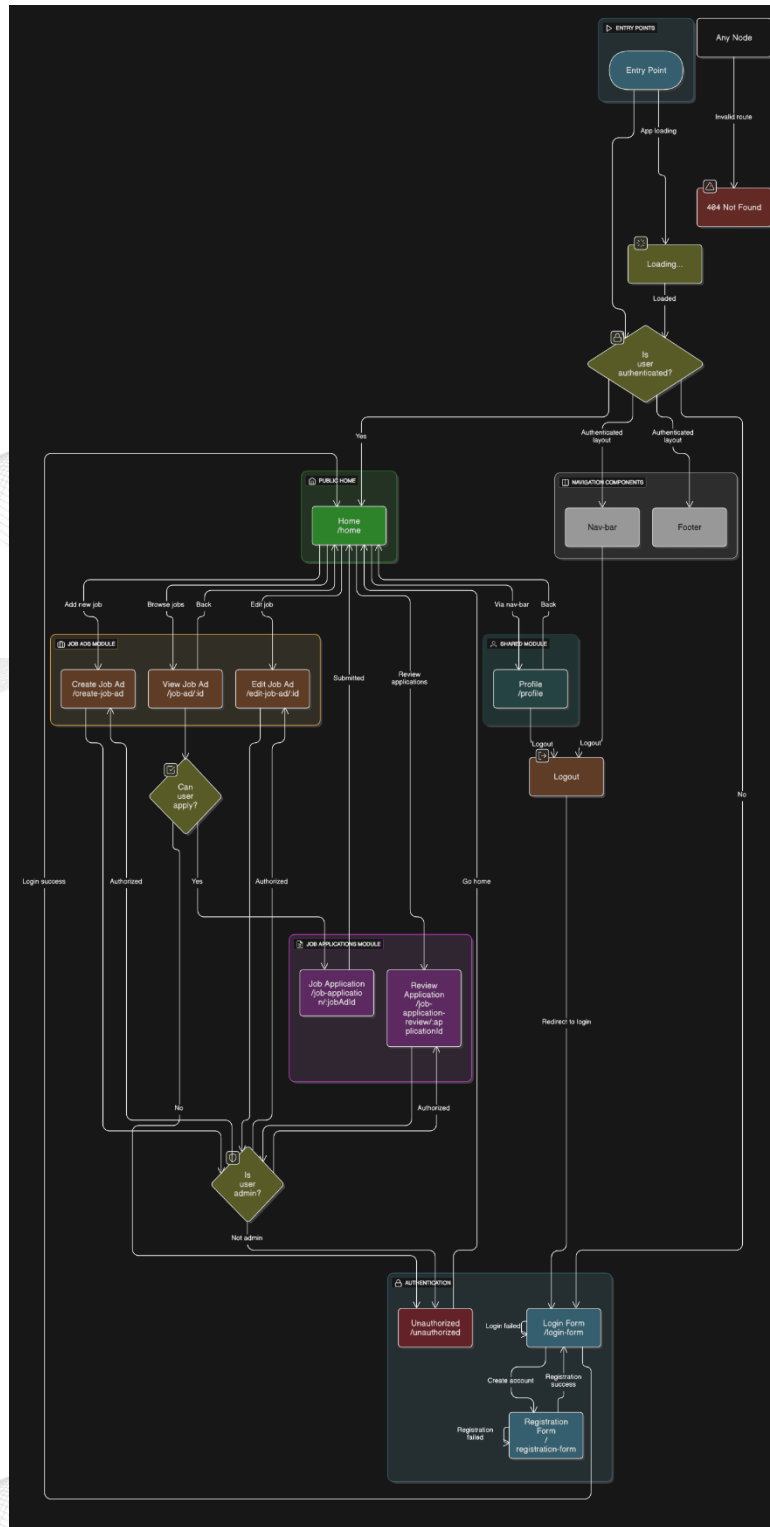
| <i>Nº</i>                    | <b>Stages</b>  |
|------------------------------|--|
| <b>1<sup>st</sup> Sprint</b> | <p><i>Backend Foundations</i></p> <p>Designed the database models, implemented 3-layer architecture, and built a fully functional API with filtering, DTOs, and services for users, job ads, and applications.</p>   |
| <b>2<sup>nd</sup> Sprint</b> | <p><i>Angular Setup &amp; Authentication</i></p> <p>Set up the Angular project, built the component structure, designed login and registration forms, and implemented authentication with role-based route guards.</p>   |
| <b>3<sup>rd</sup> Sprint</b> | <p><i>User-Side Functionality</i></p> <p>Developed user features for browsing and filtering job ads, submitting applications, viewing personal data, and ensured proper frontend validation and navigation.</p>  |
| <b>4<sup>th</sup> Sprint</b> | <p><i>Admin Features &amp; Preparation</i></p> <p>Implemented admin tools for managing job ads and reviewing applications, added GUI enhancements, and ensured full role-based access control throughout the app and that it functions properly. Enhanced the repository and created the presentation.</p> |

## IV. Used Technologies

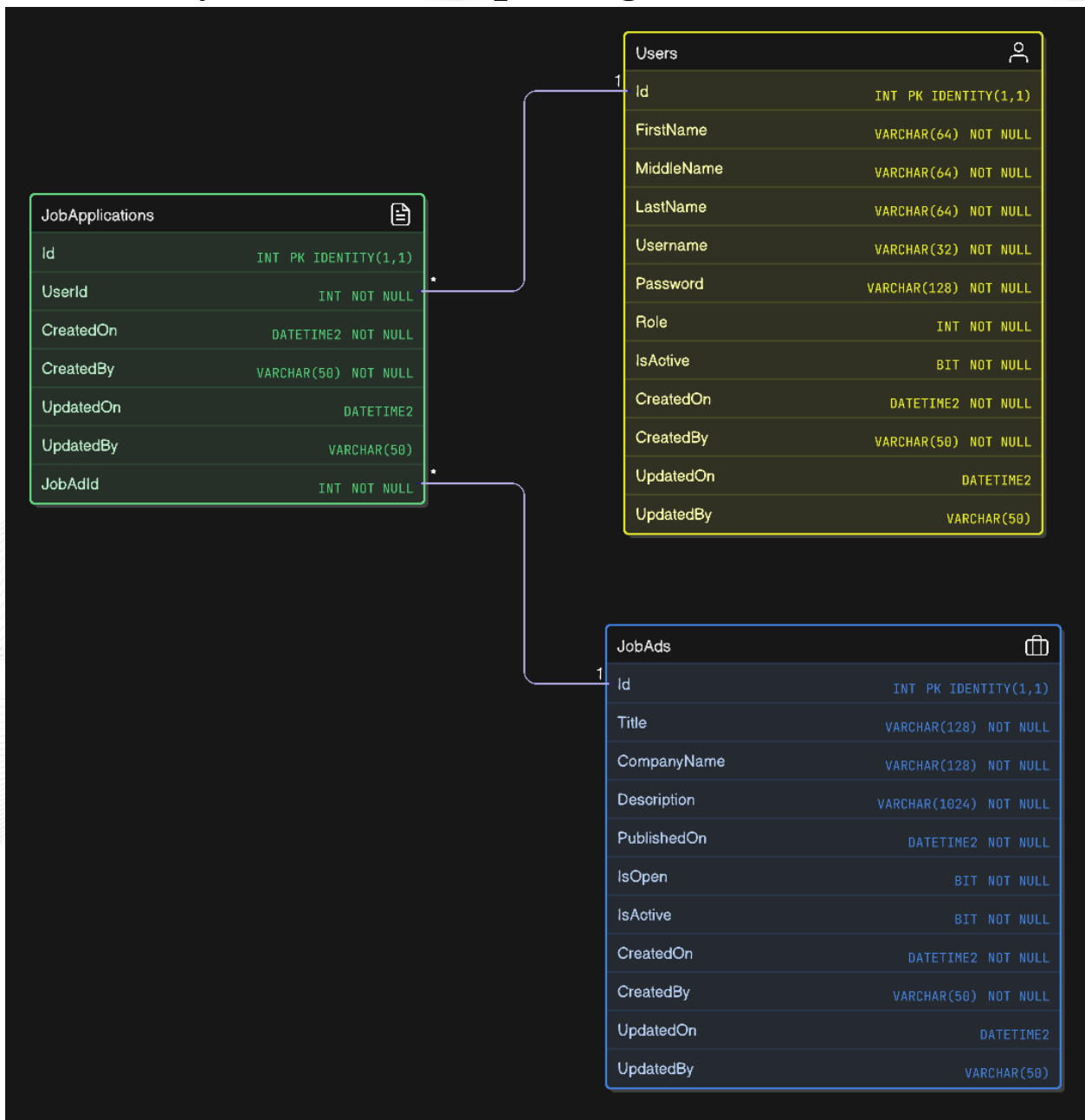




## V. Block Scheme



## VI. Entity-Relationship Diagram





## VII. Installation (to run the source code)

### 1. Prerequisites

Make sure the following software is installed on your system:

System Tools: Git, SQL Server (Express or Developer), SQL Server Management Studio (SSMS)

Backend: .NET 8 SDK

Entity Framework Core tools:

Run in terminal:

**dotnet tool install --global dotnet-ef**

Required EF Core NuGet packages (already included in the project, but install if needed):

**dotnet add package Microsoft.EntityFrameworkCore.SqlServer**

**dotnet add package Microsoft.EntityFrameworkCore.Design**

Frontend: Node.js (LTS version), Angular CLI (install globally): **npm install -g @angular/cli**

### 2. Clone the Repository

**git clone "https://github.com/codingburgas/job-tracking-BADimov21.git"**

**cd job-tracking-BADimov21**

### 3. Backend Setup (ASP.NET Core)

Navigate to the backend project folder:

**cd JobTracking.API**

After you install MSSQL Server, connect to your local instance using Windows authentication:

**(localdb)\MSSQLLocalDb**

Create the database manually in SSMS or by running this SQL command:

**CREATE DATABASE JobTracking;**

Apply migrations and generate database tables:

**dotnet ef database update**

Run the backend server:

**dotnet run**

The API should now be running at: <https://localhost:5230>

#### 4. *Frontend Setup (Angular)*

Open a new terminal and navigate to the Angular project folder:

**cd ../JobTracking.WEB**

Install dependencies:

**npm install**

Run the frontend server:

**ng serve**

The app will now be accessible at: <http://localhost:4200>

## 5. *Admin Setup*

Admin users must be manually added to the database.

Set their role to **1** in the *Users* table.

Only admin users can: Create/Edit/Delete job ads; review and manage job applications.

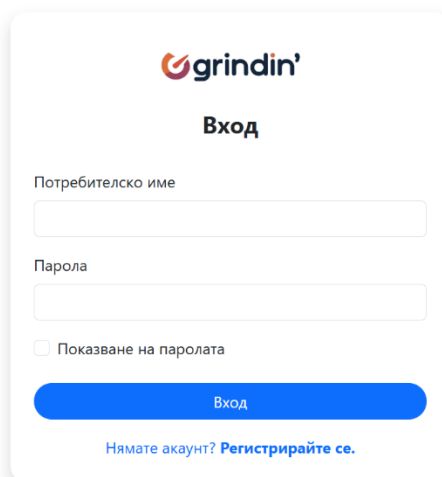
## 6. *Notes*

Role-based authentication and route protection are implemented on both frontend and backend.

Error handling for bad requests and server issues is implemented using alert messages and user feedback.

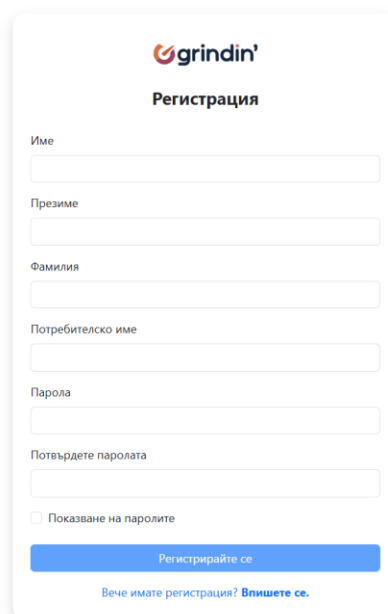
## VIII. Pictures of UI

- *Login Form*



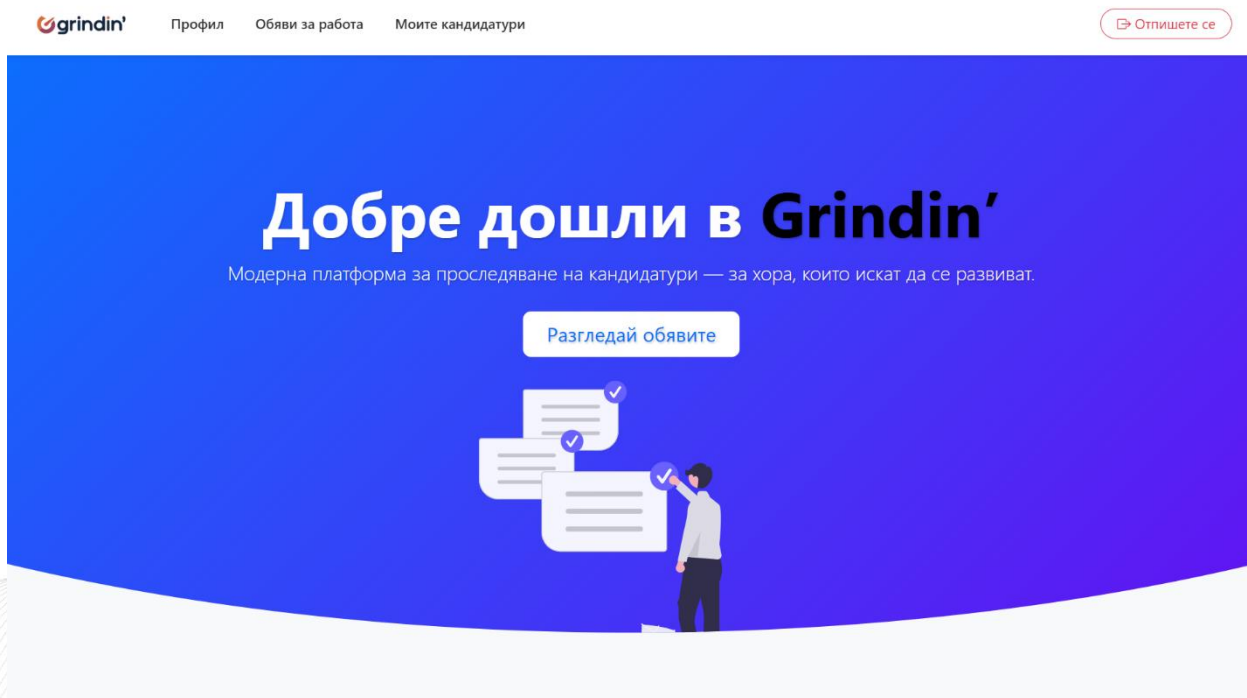
The login form is a white card with rounded corners. At the top is the 'grindin'' logo. Below it is the title 'Вход' (Login). The form contains two input fields: 'Потребителско име' (Username) and 'Парола' (Password). Below the password field is a checkbox labeled 'Показване на паролата' (Show password). A blue button labeled 'Вход' (Login) is at the bottom. Below the button is a link: 'Нямате акаунт? [Регистрирайте се.](#)' (Don't have an account? [Register here.](#))

- *Registration Form*

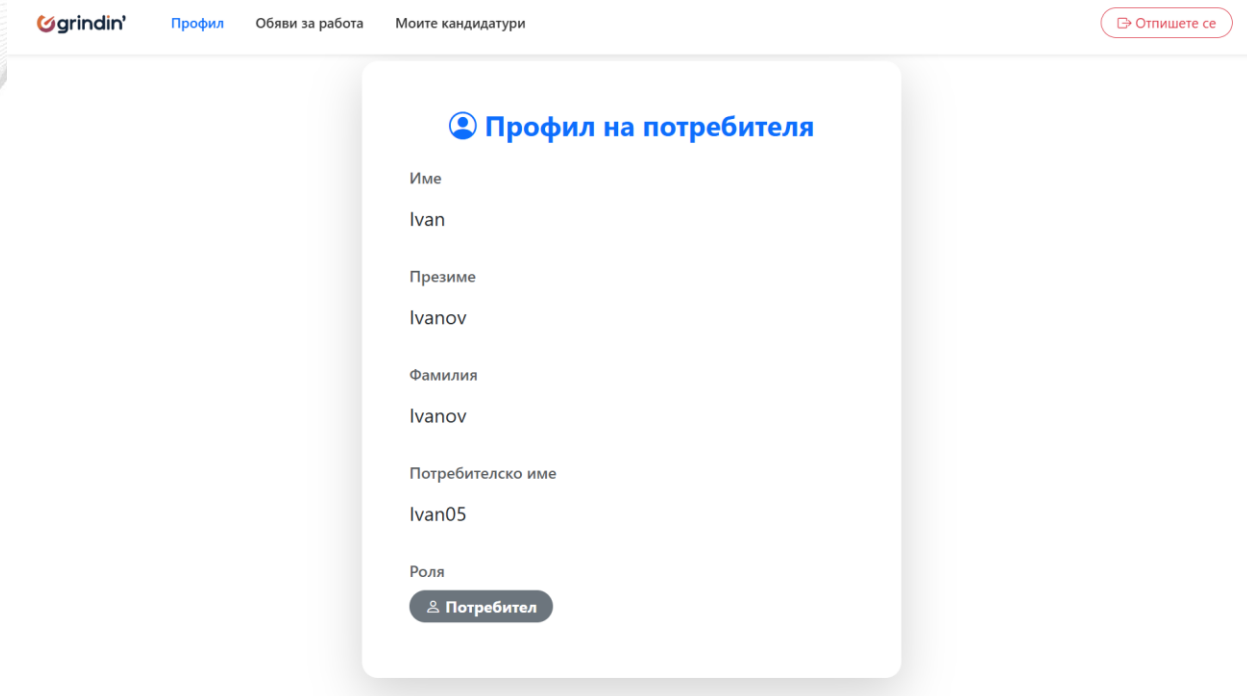


The registration form is a white card with rounded corners. At the top is the 'grindin'' logo. Below it is the title 'Регистрация' (Registration). The form contains five input fields: 'Име' (First name), 'Презиме' (Last name), 'Фамилия' (Surname), 'Потребителско име' (Username), and 'Парола' (Password). Below the password field is a checkbox labeled 'Показване на паролите' (Show passwords). A blue button labeled 'Регистрирайте се' (Register) is at the bottom. Below the button is a link: 'Вече имате регистрация? [Впишете се.](#)' (Already have an account? [Log in.](#))

- Home Page



- Profile Page



- Job Ads Page

## Обяви за работа

## Филтриране

Заглавие

Пример: Програмист

Компания

Име на фирма

Статус

Всички

Дата на публикуване

mm/dd/yyyy

Приложи филтрите

## Frontend Developer

TechNova

Responsibilities: Develop responsive UI components. Requirements: React, TypeScript, REST APIs. Conditions: Hybrid model, 25 days PTO.

Публикувана на: 27.06.2025

Активна

Кандидатствай

## Backend Developer

SoftServe

Responsibilities: Build scalable backend systems. Requirements: .NET 6, SQL Server, Dapper. Conditions: Full-time, office-based, meal vouchers.

Публикувана на: 27.06.2025

Активна

Кандидатствай

## QA Engineer

BugTrack

Responsibilities: Automate test scenarios. Requirements: Selenium, C#, Postman. Conditions: Remote-friendly, flexible hours.

Публикувана на: 27.06.2025

## DevOps Engineer

CloudStream

Responsibilities: Manage CI/CD and infrastructure. Requirements: Docker, Kubernetes, Azure. Conditions: Full remote, company laptop.

Публикувана на: 27.06.2025

- Job Applications Page

## Кандидатури за работа

## Филтриране

Статус

Всички

ID на обявата

Приложи филтрите

## Кандидатура #65

Име: Ivan Ivanov Ivanov

Потребител: Ivan05 (ID: 41)

Роля: Потребител

Обява: .NET Developer at Grindin' от Grindin' (ID: 53)

Статус: Одобрена за интервю Код: 1

## Кандидатура #66

Име: Ivan Ivanov Ivanov

Потребител: Ivan05 (ID: 41)

Роля: Потребител

Обява: Frontend Developer от TechNova (ID: 31)

Статус: Подадена Код: 0

## Кандидатура #67

Име: Ivan Ivanov Ivanov

Потребител: Ivan05 (ID: 41)

## Кандидатура #68

Име: Ivan Ivanov Ivanov

Потребител: Ivan05 (ID: 41)



## Version History

| Version | Author         | Date       | Description           |
|---------|----------------|------------|-----------------------|
| 1.0     | Bozhidar Dimov | 02.07.2025 | Initial documentation |

---

*End of documentation.*

---