Grindin’

Table of Contents

I. About me

Name	Role	Grade / Degree
Bozhidar A. Dimov	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.

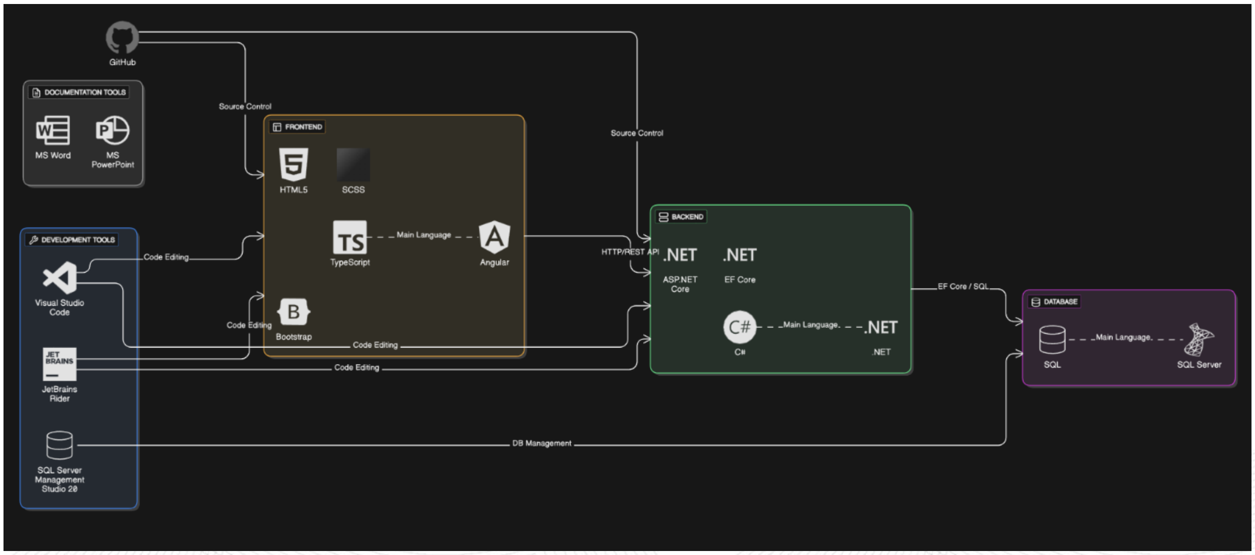
This project was developed as part of the Production Practice curriculum (2024–2025) in class 11 V at the Vocational School of Computer Programming and Innovation (VSCPI) in Burgas, Bulgaria. It serves as both a practical demonstration of software engineering skills and a full-stack web development project,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

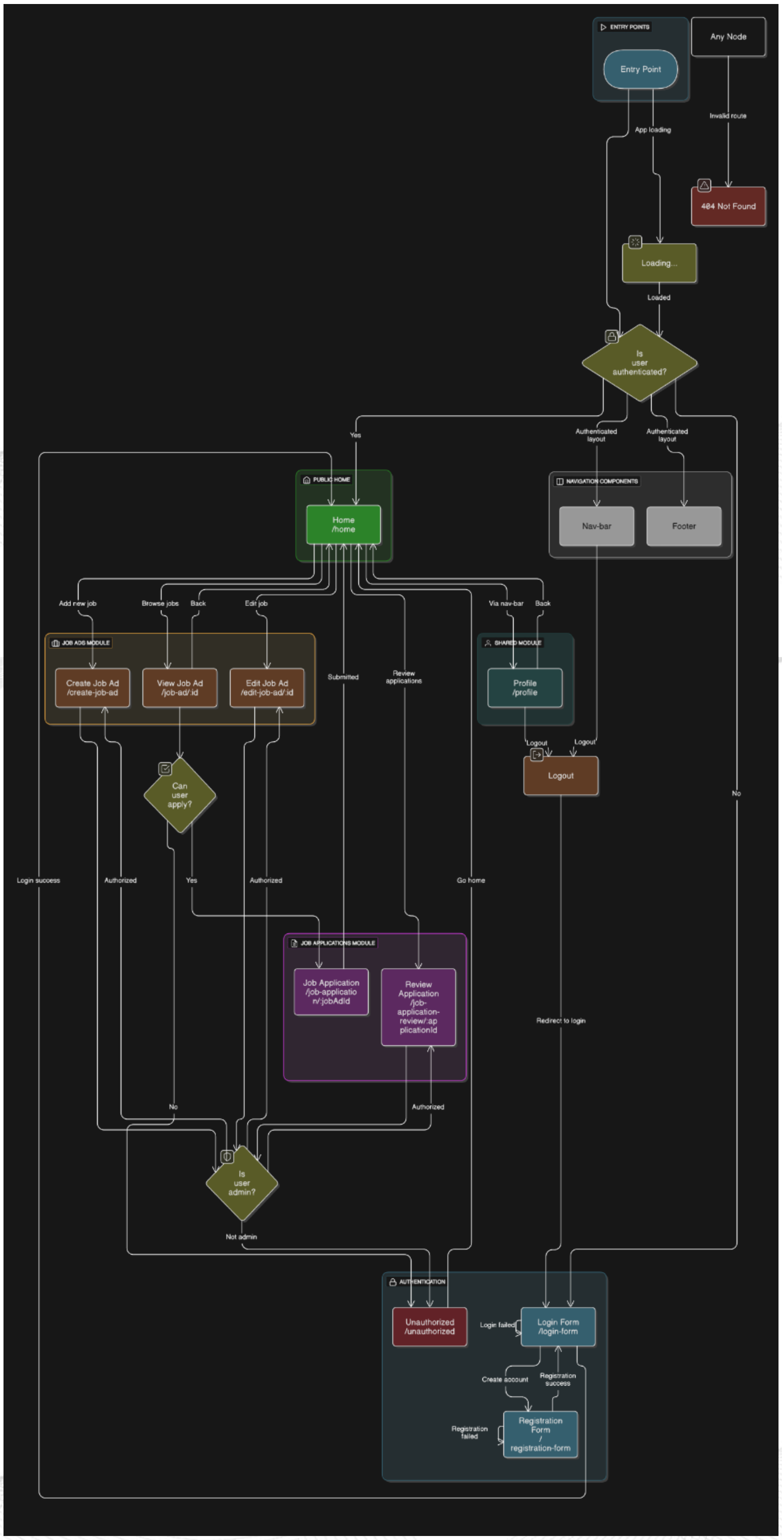
III. Stages of work

№	Stages
1st Sprint	*Backend Foundations* 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.
2nd Sprint	*Angular Setup & Authentication* Set up the Angular project, built the component structure, designed login and registration forms, and implemented authentication with role-based route guards.
3rd Sprint	*User-Side Functionality* Developed user features for browsing and filtering job ads, submitting applications, viewing personal data, and ensured proper frontend validation and navigation.
4th Sprint	*Admin Features & Preparation* 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.

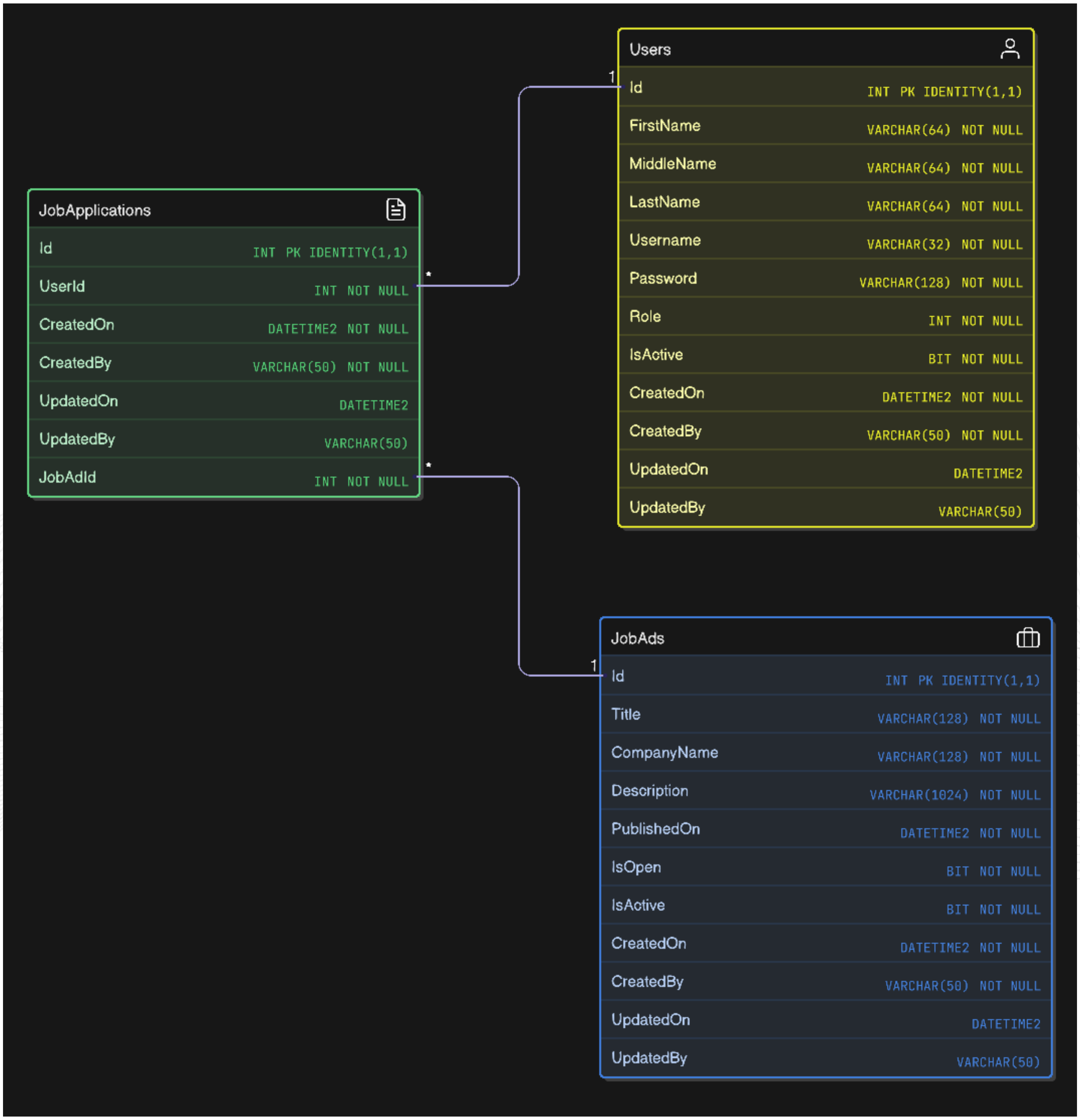
IV. Used Technologies



V. Block Scheme



VI. Entity-Relationship Diagram

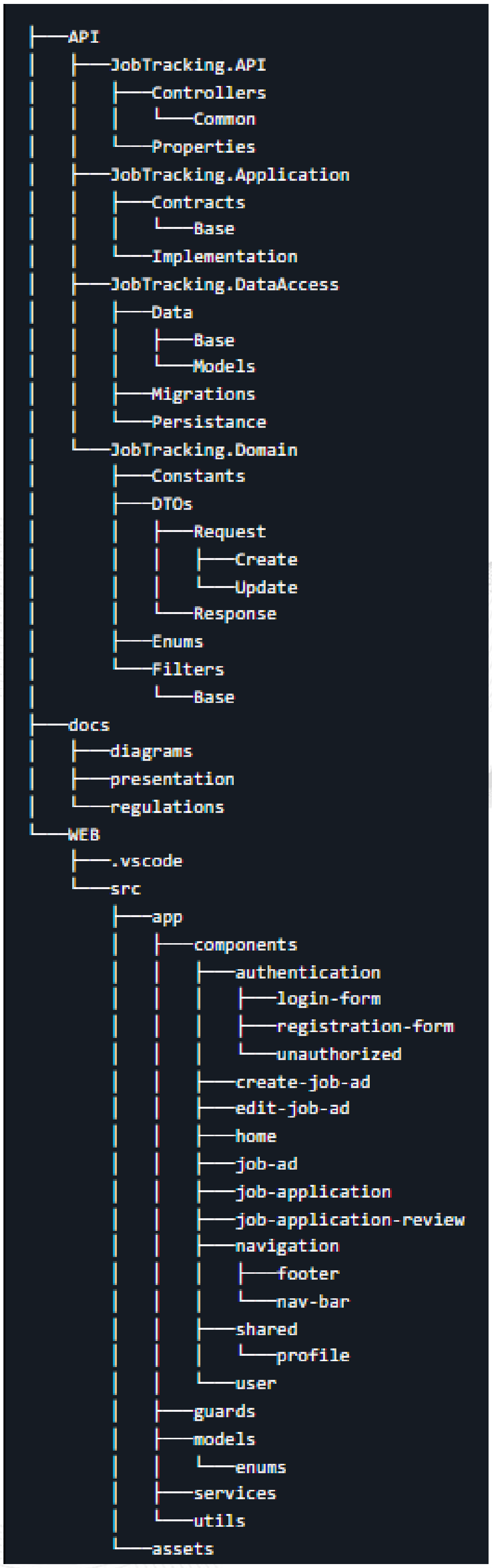


JobApplications	
Id	INT PK IDENTITY(1,1)
UserId	INT NOT NULL
CreatedOn	DATETIME2 NOT NULL
CreatedBy	VARCHAR(50) NOT NULL
UpdatedOn	DATETIME2
UpdatedBy	VARCHAR(50)
JobAdId	INT NOT NULL

Users	
Id	INT PK IDENTITY(1,1)
FirstName	VARCHAR(64) NOT NULL
MiddleName	VARCHAR(64) NOT NULL
LastName	VARCHAR(64) NOT NULL
Username	VARCHAR(32) NOT NULL
Password	VARCHAR(128) NOT NULL
Role	INT NOT NULL
IsActive	BIT NOT NULL
CreatedOn	DATETIME2 NOT NULL
CreatedBy	VARCHAR(50) NOT NULL
UpdatedOn	DATETIME2
UpdatedBy	VARCHAR(50)

JobAds	
Id	INT PK IDENTITY(1,1)
Title	VARCHAR(128) NOT NULL
CompanyName	VARCHAR(128) NOT NULL
Description	VARCHAR(1024) NOT NULL
PublishedOn	DATETIME2 NOT NULL
IsOpen	BIT NOT NULL
IsActive	BIT NOT NULL
CreatedOn	DATETIME2 NOT NULL
CreatedBy	VARCHAR(50) NOT NULL
UpdatedOn	DATETIME2
UpdatedBy	VARCHAR(50)

VII. Project Structure



```
├───API
│   ├───JobTracking.API
│   │   ├───Controllers
│   │   │   └───Common
│   │   └───Properties
│   ├───JobTracking.Application
│   │   ├───Contracts
│   │   │   └───Base
│   │   └───Implementation
│   ├───JobTracking.DataAccess
│   │   ├───Data
│   │   │   ├───Base
│   │   │   └───Models
│   │   ├───Migrations
│   │   └───Persistance
│   └───JobTracking.Domain
│       ├───Constants
│       ├───DTOs
│       │   ├───Request
│       │   │   ├───Create
│       │   │   └───Update
│       │   └───Response
│       ├───Enums
│       └───Filters
│           └───Base
├───docs
│   ├───diagrams
│   ├───presentation
│   └───regulations
└───WEB
    ├───.vscode
    └───src
        ├───app
        │   ├───components
        │   │   ├───authentication
        │   │   │   ├───login-form
        │   │   │   ├───registration-form
        │   │   │   └───unauthorized
        │   │   ├───create-job-ad
        │   │   ├───edit-job-ad
        │   │   ├───home
        │   │   ├───job-ad
        │   │   ├───job-application
        │   │   ├───job-application-review
        │   │   ├───navigation
        │   │   │   ├───footer
        │   │   │   └───nav-bar
        │   │   ├───shared
        │   │   │   └───profile
        │   │   └───user
        │   ├───guards
        │   ├───models
        │   │   └───enums
        │   ├───services
        │   └───utils
        └───assets
```

VIII. API Endpoints

Endpoint	Method	Purpose
/api/JobAd/GetById/{id}	GET	Fetch single job ad
/api/JobAd/GetAll	GET	Fetch paginated list of job ads
/api/JobAd/GetFiltered	POST	Fetch job ads with filters
/api/JobAd/Add	POST	Create a new job ad
/api/JobAd/Update/{id}	PUT	Update existing job ad
/api/JobAd/Delete/{id}	DELETE	Delete job ad
/api/JobApplication/GetById/{id}	GET	Fetch single job application
/api/JobApplication/GetAll	GET	Fetch paginated list of applications
/api/JobApplication/GetFiltered	POST	Fetch filtered job applications
/api/JobApplication/Add	POST	Submit a new job application
/api/JobApplication/Update/{id}	PUT	Update status of an application
/api/JobApplication/Delete/{id}	DELETE	Delete a job application
/api/User/GetById	GET	Fetch user by ID
/api/User/GetAll	GET	Fetch paginated list of users
/api/User/GetFiltered	POST	Fetch filtered user list
/api/User/Login	POST	Authenticate a user
/api/User/Add	POST	Register a new user
/api/User/Update/{id}	PUT	Update existing user
/api/User/Delete/{id}	DELETE	Delete a user

IX. 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.

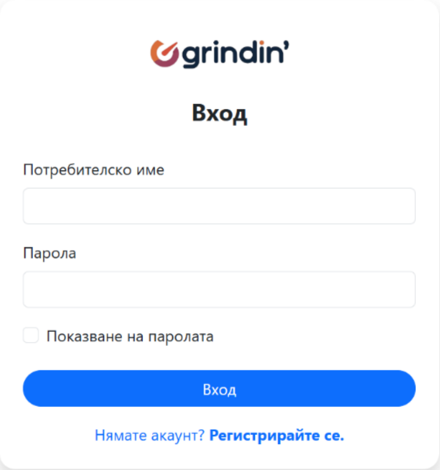
6. *Notes*

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.

X. Pictures of UI

- *Login Form*



grindin'

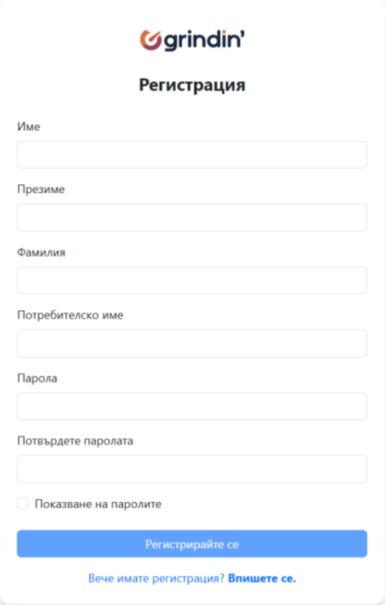
Вход

Потребителско име

Парола

□ Показване на паролата

- *Registration Form*



grindin'

Регистрация

Име

Презиме

Фамилия

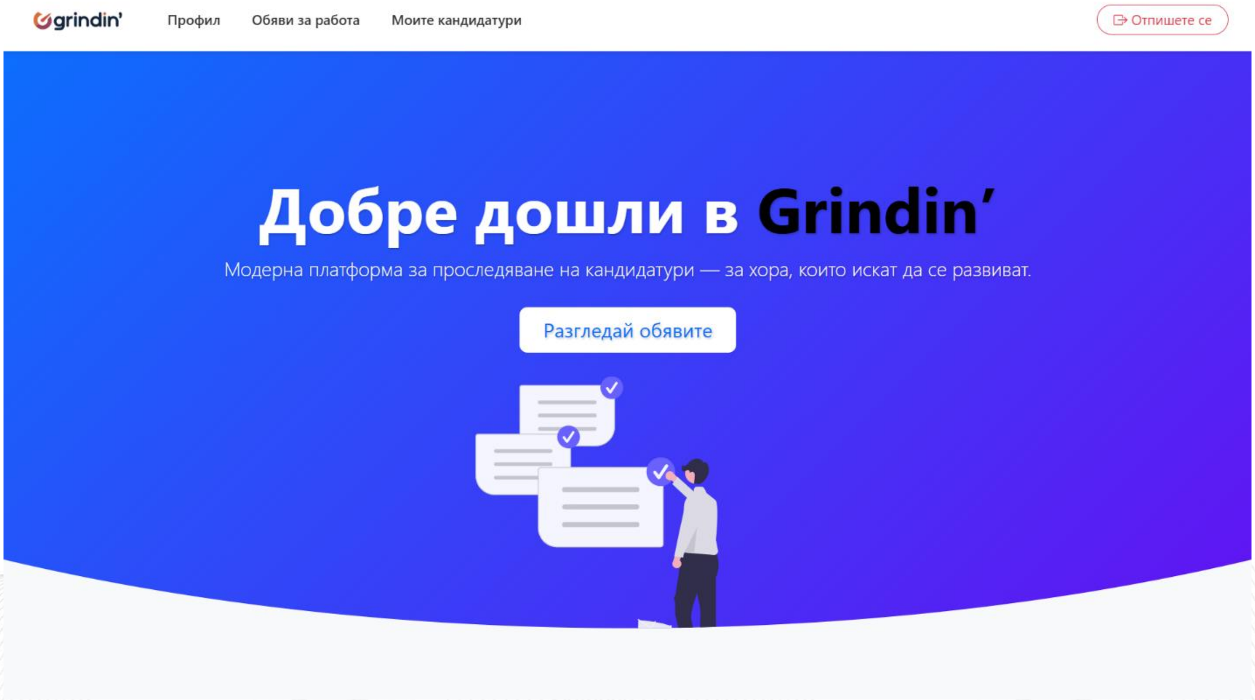
Потребителско име

Парола

Потвърдете паролата

□ Показване на паролите

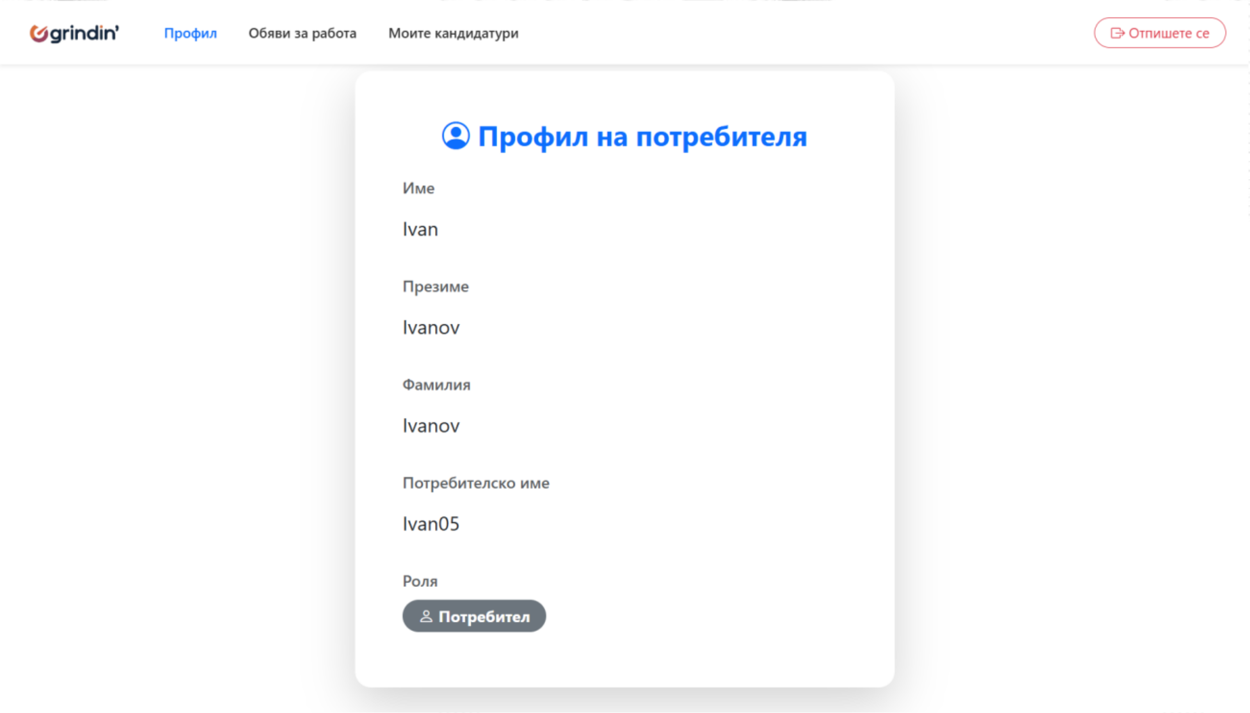
- *Home Page*



Добре дошли в Grindin'

Модерна платформа за проследяване на кандидатури — за хора, които искат да се развиват.

- *Profile Page*



Профил на потребителя

Име

Ivan

Презиме

Ivanov

Фамилия

Ivanov

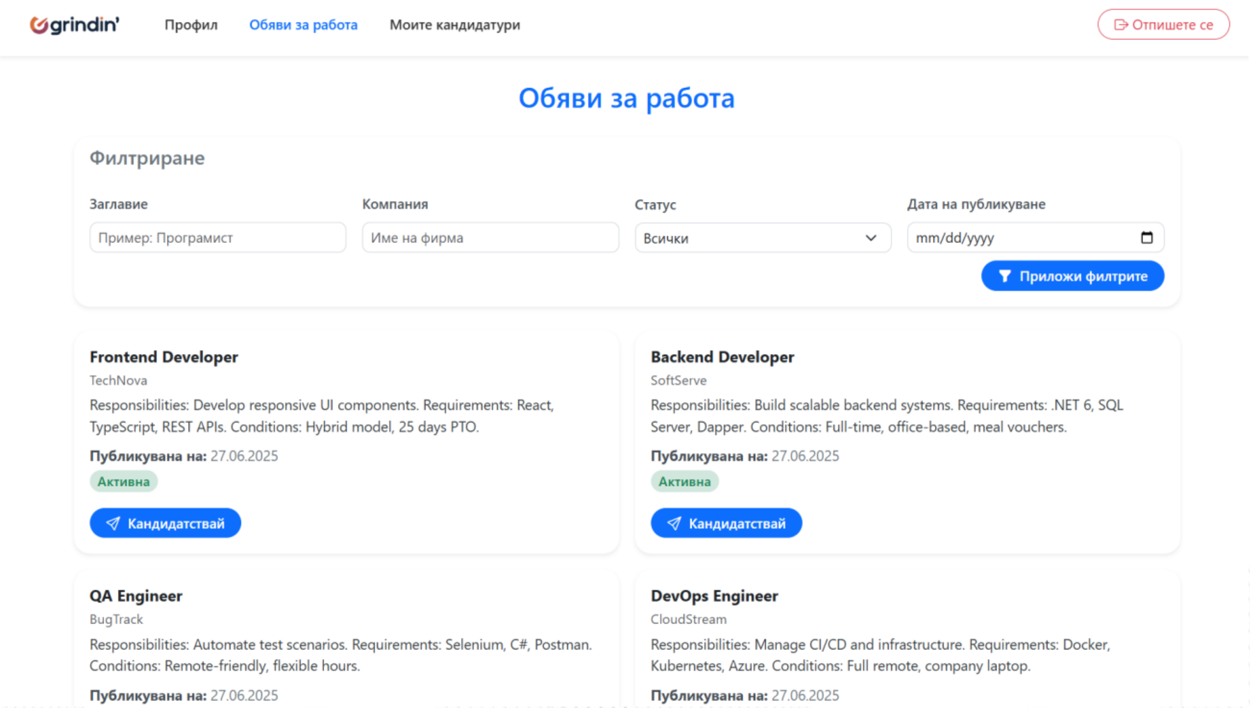
Потребителско име

Ivan05

Роля

Потребител

- *Job Ads Page*



Обяви за работа

Филтриране

Заглавие: Пример: Програмист

Компания: Име на фирма

Статус: Всички

Дата на публикуване: mm/dd/yyyy

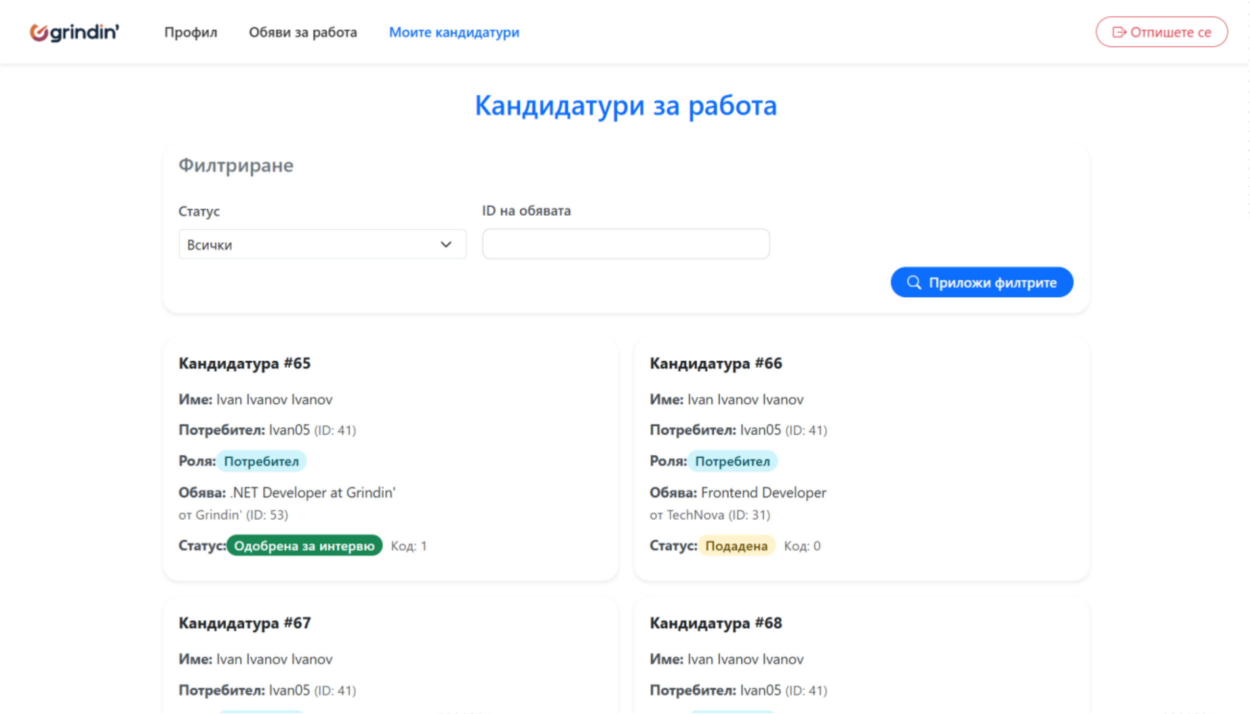
Frontend Developer
TechNova
Responsibilities: Develop responsive UI components. Requirements: React, TypeScript, REST APIs. Conditions: Hybrid model, 25 days PTO.
Публикувана на: 27.06.2025
Активна

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

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:

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.
