

# WorkZone

A web-based platform designed to help students find trusted part-time jobs while continuing their studies—without brokers or middlemen.

---

## Group Information

Role	Name	Student ID
DevOps Engineer	Anjana Madhushan	ITBIN-2313-0040
Frontend Developer	Nimanthha Vikum Kodikara	ITBIN-2313-0052
Backend Developer	Avishka Devananda	ITBIN-2313-0021

---

## Project Description

Many students currently struggle to find part-time jobs, often relying on WhatsApp groups where **brokers** reduce earnings and create trust issues.

**WorkZone** was developed to solve this problem by connecting **students and companies directly**, ensuring fair pay, transparency, and trust.

## Key Objectives

- **Centralized Platform:** A dedicated hub for student jobs.
- **Direct Interaction:** No intermediaries between students and companies.
- **Fair & Transparent:** Students receive their full earnings.
- **Secure System:** A safe environment for all users.

---

## Live Deployment

**Live URL:** <https://work-zone-orcin.vercel.app/>

## Discord

**Discord Link:** <https://discordapp.com/channels/1452031123772407992/1462883918498889810>

The application is deployed and accessible via the link above.

---

## Main Features

- **Secure Authentication:** Robust login systems for both Students and Companies.
- **Job Management:** Easy job posting and management for companies.
- **Student Applications:** Seamless job application process for students.
- **Responsive UI:** Modern, mobile-friendly design built with **Tailwind CSS**.
- **Broker-Free:** Direct connections ensuring maximum earnings for students.

## Technologies Used

- **Frontend:** HTML5, CSS3, JavaScript, Tailwind CSS
  - **Backend:** Node.js, Express.js, MongoDB
  - **DevOps:** GitHub Actions, Shell Scripting, Vercel/Render
  - **Tools:** Git, VS Code, Postman
- 

## Branching Strategy

We implemented the following branching strategy to ensure code quality:

- **main - Production Branch:** Auto-deployed to live server.
- **develop - Integration Branch:** All features are merged here first.
- **feature/\* - Feature Branches:** Individual branches for each developer.

**Workflow:** Feature -> Develop (Testing) -> Main (Production)

---

## Individual Contributions

### Anjana Madhushan (DevOps Engineer)

- Initialized the repository and configured `.gitignore`.
- Implemented **GitHub Actions** for CI/CD pipelines (`ci/cd.yml`, `main.yml`).
- Managed deployment configurations on **Render/Vercel**.
- Wrote shell scripts for automation and managed environment variables.
- Resolved merge conflicts during the integration phase.

### Nimantha Vikum Kodikara (Frontend Developer)

- Designed the responsive UI using **Tailwind CSS**.
- Developed the Student and Company dashboard layouts.
- Implemented frontend validation for Login/Register forms.
- Integrated Frontend with Backend APIs using Axios.
- Improved mobile responsiveness and UX.

### Avishka Devananda (Backend Developer)

- Designed the Backend architecture and Database schema.
  - Implemented Authentication APIs (Login/Register/Google Auth).
  - Created Job Management and User Profile APIs.
  - Configured MongoDB database connections.
  - Handled API security and error handling.
- 

## Setup Instructions

### Prerequisites

- **Node.js** (v18 or higher)
- **Git** installed

## Installation

1. **Clone the repository** `bash` `git clone [https://github.com/AnjanaMadhushanaj/WorkZone.git] (ht`  
`cd workzone`
2. **Install Dependencies**

```
# Install Backend
cd backend
npm install

# Install Frontend
cd ../frontend
npm install
```

3. **Run the Application**

```
# Run Backend (Port 5000)
npm run dev

# Run Frontend (Port 5173/3000)
npm run dev
```

## Deployment Process (CI/CD)

Our project uses **GitHub Actions** for automation: 1. **CI Pipeline:** Triggers on pull requests to **develop** or **main**. It installs dependencies, builds the project, and runs tests. 2. **Deployment:** Triggers **only** when changes are merged into **main**. It automatically deploys the latest code to the production server.

## Challenges Faced

- **CORS Errors:** We faced issues connecting Frontend to Backend due to CORS policies, resolved by configuring the `cors` package in Express.
  - **Merge Conflicts:** Encountered conflicts in `package.json` when merging branches, resolved by manual code review.
- 

## Build Status

Built for Systems Administration & Maintenance Assignment (2026)