DESIGN THINKING II

LABOUR-EASE

SUBMITTED TO:

Ms. Anuradha Purohit

Ms. Ritambhara Patidar

SUBMITTED BY:

Prateek Khandekar(0801CS221111)

Purvi Porwal(0801CS221114)

Rishita Sharma(0801CS221120)

Shiwang Sharma(0801CS221134)

Tanya Agrawal (0801CS221172)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

2024-2025

INDEX

Requirement Analysis

We have a set of users, each user has an ID (unique identifier), username, password, email, and role, which can either be 'shopkeeper' or 'labour'. The role defines their functionality in the application.

Each shopkeeper has an ID, shop name, shop address, and phone number. Each shopkeeper can post jobs that require specific skills.

Each labour has an ID, name, phone number, address, and experience. Each labour can list the skills they possess and search for jobs that match these skills.

We maintain a set of skills that can be assigned to labours. These skills are also linked to job postings, where a shopkeeper specifies which skills are required for a given job.

Shopkeepers can post job openings specifying a job title and description. Each job posting requires certain skills, and these skills are matched with the labour's skill sets.

For each job posting, shopkeepers can view and hire labours. Once a labour is assigned to a job, we store information about which shopkeeper, labour, and job posting are linked, along with the starting time of the work.

For each **job posting**, we need to keep track of the following:

- The date on which the job was posted
- The description of the job
- The skills required for the job
- The labours who applied for the job and their assigned status

For each **labour** who works for a shopkeeper, we need to track the start time when they begin their work on a specific job posting.

Constraints

- Every user must have a role, either 'shopkeeper' or 'labour'.
- Every job posting must be linked to a shopkeeper.
- Each labour must belong to one or more skills, and each job posting must require one or more skills.
- Referential integrity is enforced by foreign keys, ensuring that when a shopkeeper, labour, or job posting is deleted, related records are automatically removed.

Objectives

- Provide an interface for shopkeepers to post jobs.
- Enable labours to register their skills and find job postings matching their qualifications.
- Store and manage information about users, job postings, skills.

Functional Requirements

- **User Management**:
  - Register users as either shopkeepers or labours.
  - Allow users to update profiles.
- **Shopkeeper Management**:
  - Shopkeepers can create job postings.
  - Manage shop details (shop name, address, phone).
  - View and manage job postings.
- **Labour Management**:
  - Labours can register, list their skills, and apply for job postings.
  - Manage labour details (name, phone, address, experience).
  - View available job postings and apply to relevant ones.
- **Job Posting**:
  - Shopkeepers can post jobs specifying the required skills.
  - Labours can search for job postings based on their skills.
  - Shopkeeper can track which labours are assigned to which job postings.
- **Skill Management**:
  - Labours can add or remove their skills.
  - Shopkeepers can specify required skills for job postings.

Non-Functional Requirements

- **Performance**: The database should be optimized for quick retrieval of job postings and user information.
- **Security**: Passwords must be stored securely. Sensitive user information such as emails should be protected.
- **Scalability**: The system should handle a large number of job postings and user registrations.
- **Data Integrity**: Enforce referential integrity using foreign keys. Ensure that job postings and skills are properly linked.

DDL

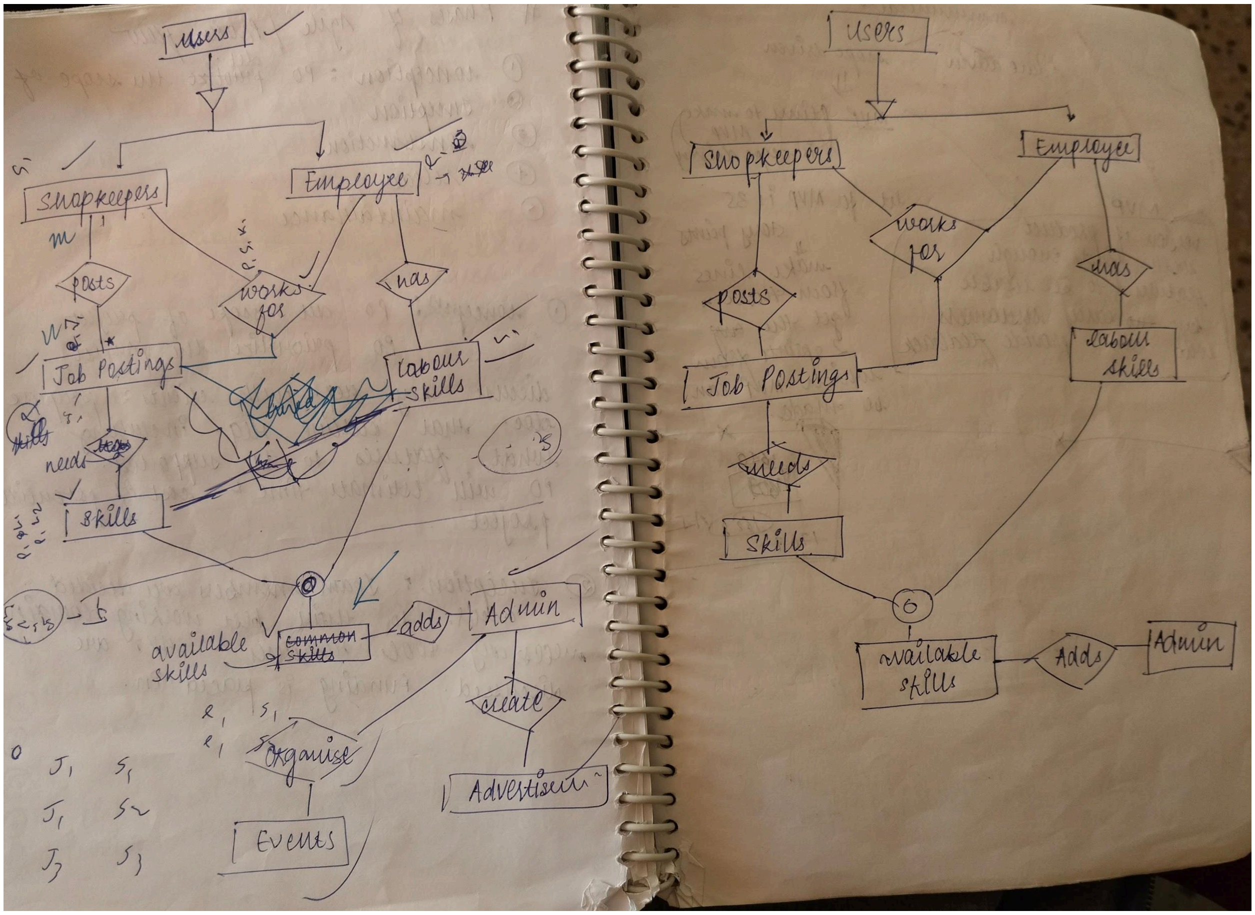
```
--  User table
CREATE TABLE users (
   id SERIAL PRIMARY KEY,
   username VARCHAR(50) NOT NULL,
   password VARCHAR(255) NOT NULL,
   email VARCHAR(100) NOT NULL,
   role VARCHAR(20) NOT NULL CHECK (role IN ('shopkeeper', 'labour'))
);

--  Shopkeeper table
CREATE TABLE shopkeepers (
   id INTEGER PRIMARY KEY,
   shop_name VARCHAR(100) NOT NULL,
   shop_address TEXT NOT NULL,
   shop_phone VARCHAR(20) NOT NULL,
   created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
   updated_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
   FOREIGN KEY (id) REFERENCES users(id) ON DELETE CASCADE
);

--  Labour table
CREATE TABLE labours (
   id INTEGER PRIMARY KEY,
   name VARCHAR(50) NOT NULL,
   phone VARCHAR(20) NOT NULL,
   address TEXT NOT NULL,
   experience TEXT NOT NULL,
   created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
   updated_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
   FOREIGN KEY (id) REFERENCES users(id) ON DELETE CASCADE
);

--  Skills table
CREATE TABLE skills (
   id SERIAL PRIMARY KEY,
   name VARCHAR(50) NOT NULL,
   created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
   updated_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
```

```sql
);

--  Labour_has_Skills table
CREATE TABLE labour_has_skills (
   labour_id INTEGER NOT NULL,
   skill_id INTEGER NOT NULL,
   PRIMARY KEY (labour_id, skill_id),
   FOREIGN KEY (labour_id) REFERENCES labours(id) ON DELETE CASCADE,
   FOREIGN KEY (skill_id) REFERENCES skills(id) ON DELETE CASCADE
);

--  Job Posting table
CREATE TABLE job_postings (
   id SERIAL PRIMARY KEY,
   shopkeeper_id INTEGER NOT NULL,
   title VARCHAR(100) NOT NULL,
   description TEXT NOT NULL,
   created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
   updated_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
   FOREIGN KEY (shopkeeper_id) REFERENCES shopkeepers(id) ON DELETE CASCADE
);

--  Posting_Needs_Skills table
CREATE TABLE posting_needs_skills (
   job_posting_id INTEGER NOT NULL,
   skill_id INTEGER NOT NULL,
   PRIMARY KEY (job_posting_id, skill_id),
   FOREIGN KEY (job_posting_id) REFERENCES job_postings(id) ON DELETE CASCADE,
   FOREIGN KEY (skill_id) REFERENCES skills(id) ON DELETE CASCADE
);

--  Works_For table
CREATE TABLE works_for (
   shopkeeper_id INTEGER,
   labour_id INTEGER,
   job_posting_id INTEGER,
   starting_time TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
   PRIMARY KEY (shopkeeper_id, labour_id, job_posting_id),
   FOREIGN KEY (shopkeeper_id) REFERENCES shopkeepers(id) ON DELETE CASCADE,
   FOREIGN KEY (labour_id) REFERENCES labours(id) ON DELETE CASCADE,
   FOREIGN KEY (job_posting_id) REFERENCES job_postings(id) ON DELETE CASCADE
);
```

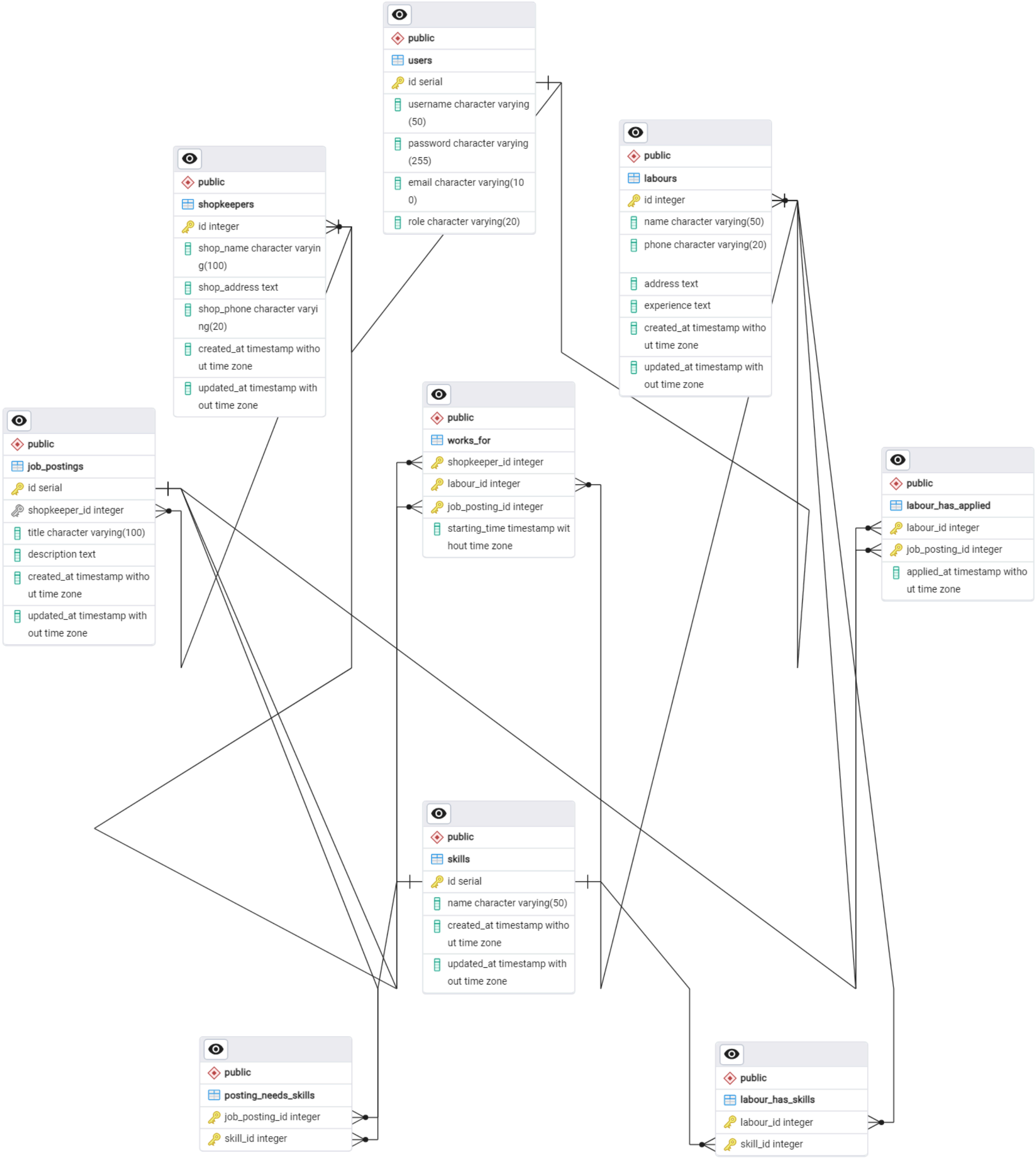
– Applied table

```
CREATE TABLE labour_has_applied (
   labour_id INTEGER NOT NULL,
   job_posting_id INTEGER NOT NULL,
   applied_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
   PRIMARY KEY (labour_id, job_posting_id),
   FOREIGN KEY (labour_id) REFERENCES labours(id) ON DELETE CASCADE,
   FOREIGN KEY (job_posting_id) REFERENCES job_postings(id) ON DELETE CASCADE
);
```

ER diagram

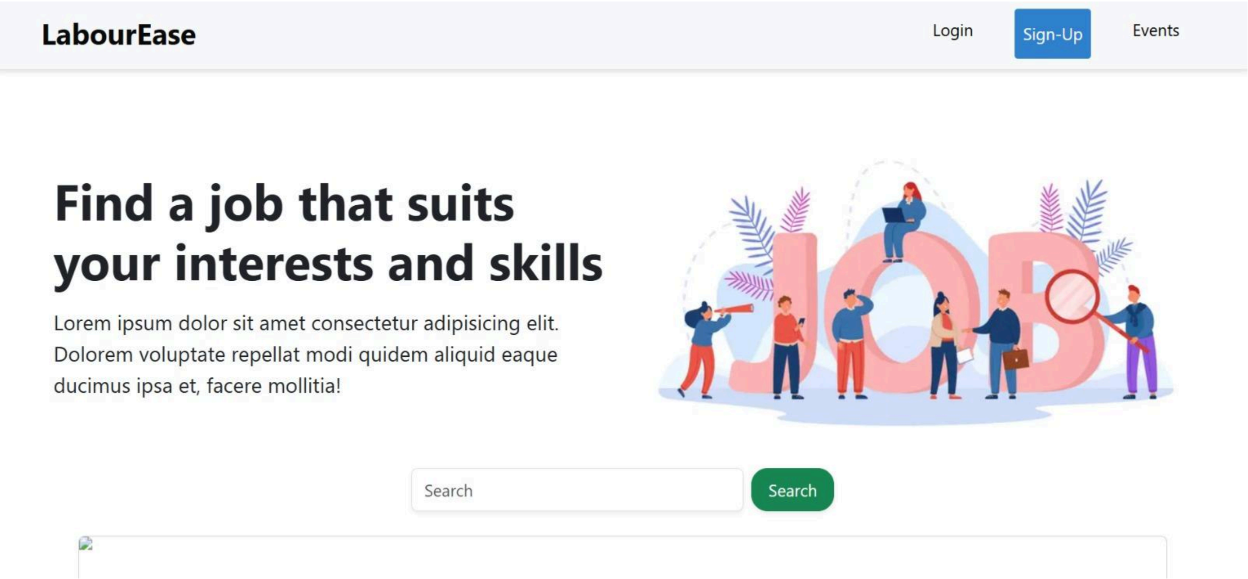

Users
Shopkeepers
Employee
posts
works for
has
Job Postings
Labour skills
needs
Skills
~~common skills~~
available skills
adds
Admin
create
Organise
Events
Advertisement
Users
Shopkeepers
Employee
posts
works for
has
Job Postings
Labour skills
needs
Skills
available skills
Adds
Admin

Relational Model


public
users
id serial
username character varying (50)
password character varying (255)
email character varying(100)
role character varying(20)
public
shopkeepers
id integer
shop_name character varying(100)
shop_address text
shop_phone character varying(20)
created_at timestamp without time zone
updated_at timestamp without time zone
public
labours
id integer
name character varying(50)
phone character varying(20)
address text
experience text
created_at timestamp without time zone
updated_at timestamp without time zone
public
job_postings
id serial
shopkeeper_id integer
title character varying(100)
description text
created_at timestamp without time zone
updated_at timestamp without time zone
public
works_for
shopkeeper_id integer
labour_id integer
job_posting_id integer
starting_time timestamp without time zone
public
labour_has_applied
labour_id integer
job_posting_id integer
applied_at timestamp without time zone
public
skills
id serial
name character varying(50)
created_at timestamp without time zone
updated_at timestamp without time zone
public
posting_needs_skills
job_posting_id integer
skill_id integer
public
labour_has_skills
labour_id integer
skill_id integer

Implementation

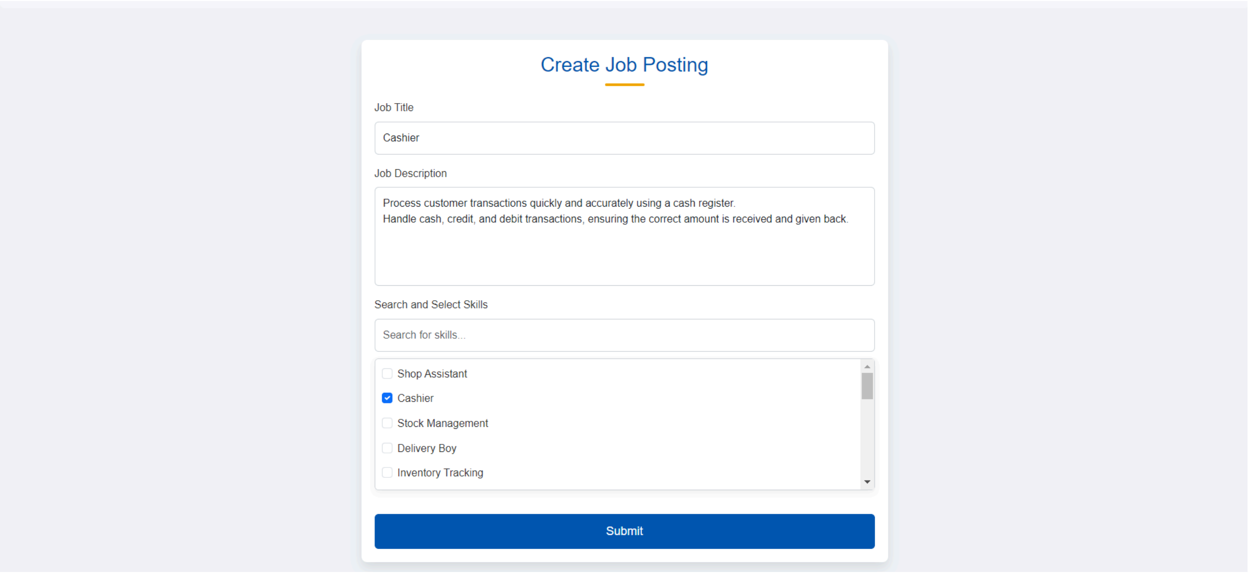
Home Page



Find a job that suits your interests and skills

Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolorem voluptate repellat modi quidem aliquid eaque ducimus ipsa et, facere mollitia!

Job Posting



Job Title

Cashier

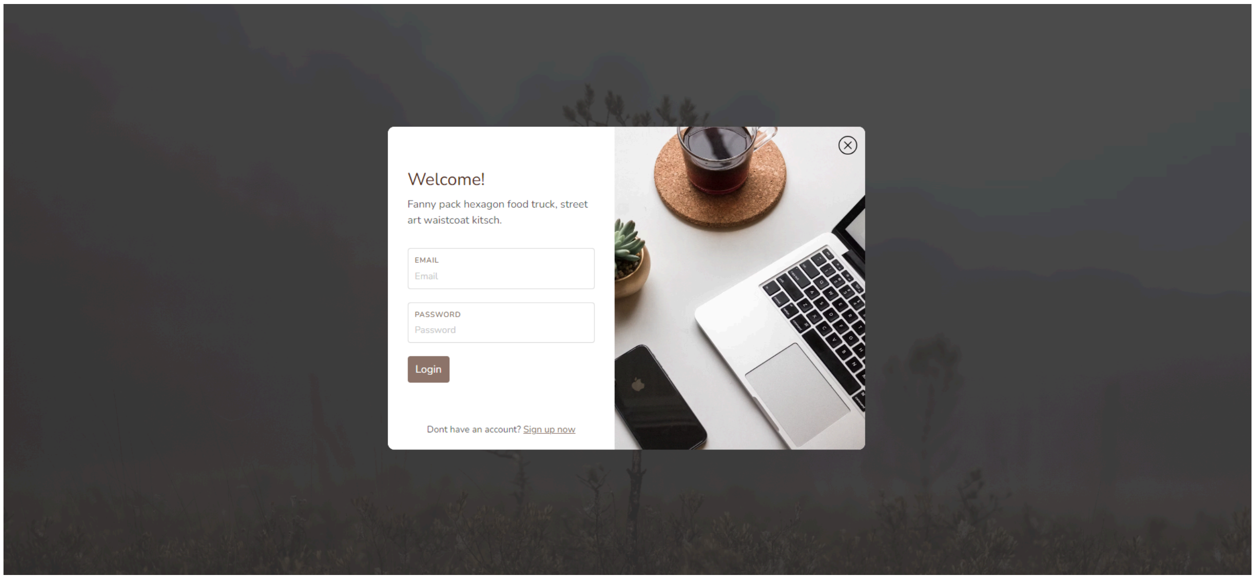
Job Description

Process customer transactions quickly and accurately using a cash register.
Handle cash, credit, and debit transactions, ensuring the correct amount is received and given back.

Search and Select Skills

- □ Shop Assistant
- ☑ Cashier
- □ Stock Management
- □ Delivery Boy
- □ Inventory Tracking

Login



Welcome!

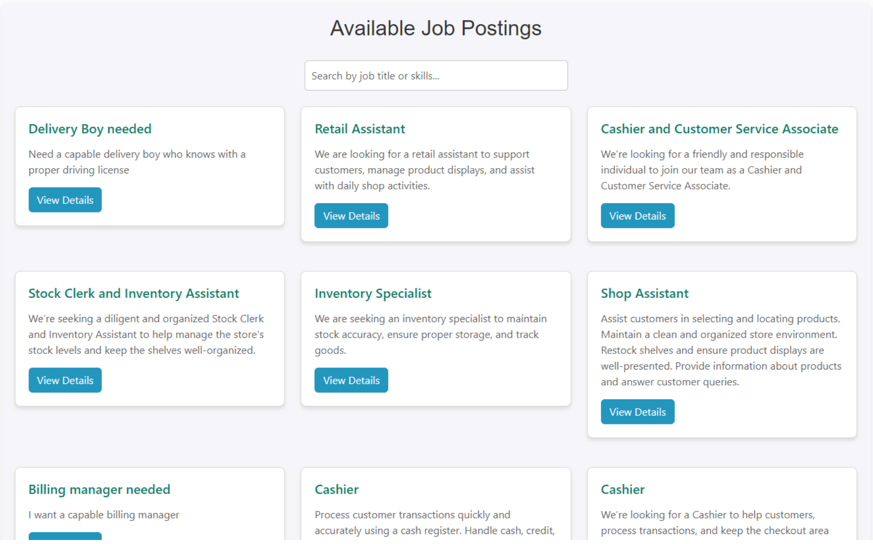
Fanny pack hexagon food truck, street art waistcoat kitsch.

EMAIL
Email

PASSWORD
Password

Dont have an account? Sign up now

Available Jobs



Available Job Postings

Search by job title or skills...

Delivery Boy needed

Need a capable delivery boy who knows with a proper driving license

Retail Assistant

We are looking for a retail assistant to support customers, manage product displays, and assist with daily shop activities.

Cashier and Customer Service Associate

We're looking for a friendly and responsible individual to join our team as a Cashier and Customer Service Associate.

Stock Clerk and Inventory Assistant

We're seeking a diligent and organized Stock Clerk and Inventory Assistant to help manage the store's stock levels and keep the shelves well-organized.

Inventory Specialist

We are seeking an inventory specialist to maintain stock accuracy, ensure proper storage, and track goods.

Shop Assistant

Assist customers in selecting and locating products. Maintain a clean and organized store environment. Restock shelves and ensure product displays are well-presented. Provide information about products and answer customer queries.

Billing manager needed

I want a capable billing manager

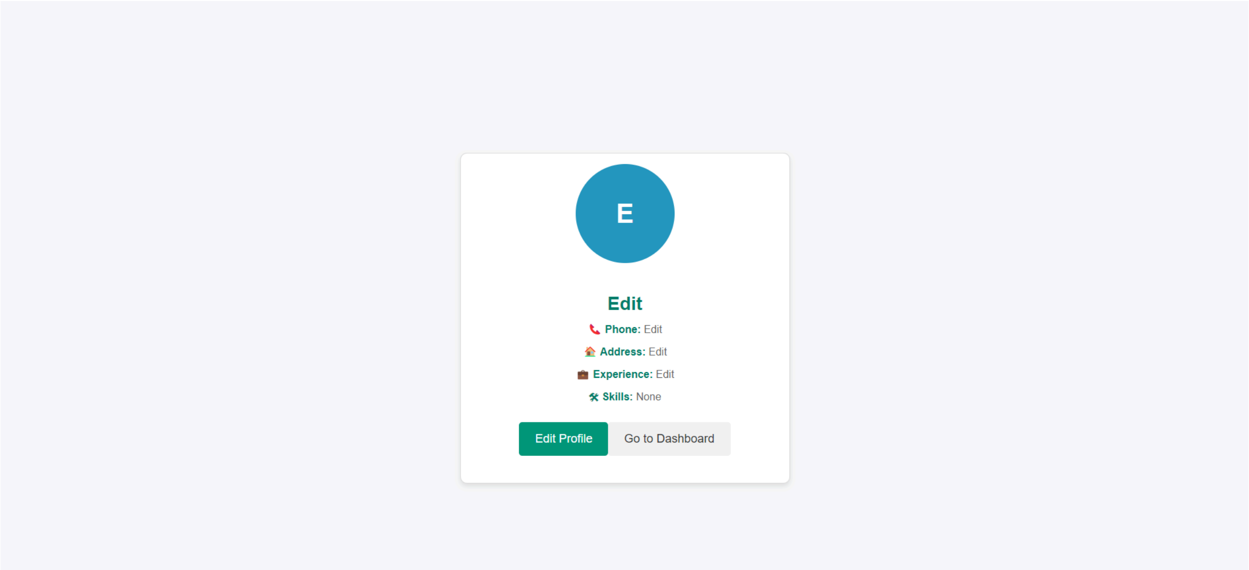
Cashier

Process customer transactions quickly and accurately using a cash register. Handle cash, credit,

Cashier

We're looking for a Cashier to help customers, process transactions, and keep the checkout area

Edit Profile



Edit

📞 **Phone:** Edit

🏠 **Address:** Edit

💼 **Experience:** Edit

🛠 **Skills:** None