

# ReUselt - Gamified Recycling App

---

## Overview

---

ReUselt is a mobile application designed to gamify the recycling process, making environmental sustainability engaging and rewarding. The app leverages AI-powered waste identification to help users correctly sort and recycle waste materials, earning points and achievements along the way.

## Features

---

- **AI-Powered Waste Identification:** Uses machine learning models to identify waste types from photos
- **Gamification System:** Earn points, unlock achievements, and compete with friends
- **Community Features:** Share recycling tips and connect with like-minded users
- **Location-Based Services:** Find nearby recycling centers and events
- **Progress Tracking:** Monitor personal and community recycling impact

## Technology Stack

---

### Backend

- **Framework:** NestJS with GraphQL API
- **Database:** MongoDB with Prisma ORM
- **Authentication:** JWT-based authentication
- **AI/ML:** TensorFlow Lite for on-device inference, Ollama LLM integration

### Frontend

- **Framework:** React Native with Expo
- **Styling:** NativeWind (Tailwind CSS for React Native)
- **State Management:** Apollo Client for server state, Zustand for global UI state
- **Navigation:** Expo Router

### Development Tools

- **Version Control:** Git with GitHub
- **Package Management:** pnpm workspaces
- **Containerization:** Docker Compose
- **Testing:** Jest for unit and integration tests

# Project Structure

---

```
Raut-PranavVirendra_4243687_PSE/  
├── 01-Project-management/  
├── 02-Requirements/  
├── 03-Architecture-documentation/  
├── 04-Implementation/  
│   └── Raut-PranavVirendra_4243687_PSE_Submission_Code.zip  
├── README.md (this file)  
└── github-link.txt
```

## Installation and Setup

---

### Prerequisites

- Node.js (v18 or higher)
- pnpm
- Docker and Docker Compose
- Python 3.8+ (for ML training)
- Expo CLI (for mobile development)

## Backend Setup

1. Navigate to the backend directory:

```
cd apps/backend
```

2. Install dependencies:

```
pnpm install
```

3. Set up environment variables:

```
cp .env.example .env  
# Edit .env with your configuration
```

4. Start MongoDB and Redis:

```
docker-compose up -d mongo redis
```

5. Run database migrations:

```
pnpm prisma:generate  
pnpm prisma:migrate:dev
```

6. Start the development server:

```
pnpm run start:dev
```

## Mobile App Setup

1. Navigate to the mobile directory:

```
cd apps/mobile
```

2. Install dependencies:

```
pnpm install
```

3. Set up environment variables:

```
cp .env.example .env  
# Edit .env with your configuration
```

4. Generate GraphQL types:

```
pnpm codegen
```

5. Start the Expo development server:

```
pnpm run start
```

## ML Training Setup (Optional)

1. Navigate to the ML training directory:

```
cd apps/ml-training
```

2. Create virtual environment:

```
python -m venv .venv  
source .venv/bin/activate
```

3. Install dependencies:

```
pip install -r requirements.txt
```

4. Run training scripts as needed.

# Usage

---

## Running the Application

1. Ensure backend services are running
2. Start the mobile app via Expo
3. Scan waste items to identify recyclable materials
4. Earn points and track your environmental impact

## Development

- Backend API available at `http://localhost:3000/graphql`
- Mobile app runs on Expo development client
- Use `pnpm test` to run test suites

## Contributing

---

This project follows a backend-first development workflow:

1. Update Prisma schema
2. Implement GraphQL resolvers
3. Add tests
4. Update frontend components
5. Run code generation

## License

---

This project is developed as part of the IU International University of Applied Sciences Software Engineering course portfolio.

## Contact

---

Pranav Virendra Raut

Student ID: 4243687

Course: Project Software Engineering (DLMCSPSE01)