# Project Report: BarCodeScanApp

#### 1. Project Title

BarCodeScanApp - A Full Stack Barcode Scanning and Batch Management Application

#### 2. Objective

To develop a mobile application that enables users to scan barcodes using their device camera, store batch-related information securely in a MongoDB database via a Node.js backend, and provide a user-friendly interface to view and manage all scanned entries.

#### 3. Motivation

Barcode systems are widely used in inventory management, warehouse tracking, and educational setups for identification. A mobile-based solution that offers scanning, saving, and reviewing scanned data in one place can significantly simplify manual entry and reduce errors.

### 4. Tech Stack Used

# Frontend:

- React Native (Expo)
- React Navigation
- Expo Camera

#### Backend:

- Node.js
- Express.js
- MongoDB + Mongoose
- dotenv, CORS

#### 5. System Architecture

### Client (Mobile App):

- Scans barcodes using the camera
- Sends batch data to backend using HTTP requests
- Retrieves list of saved batches and displays in a table view

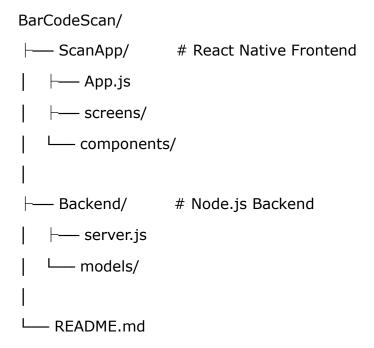
#### Server (Node + Express):

- Receives scanned data through POST request
- Saves entries in MongoDB
- Provides all entries on GET request

# Database (MongoDB):

Stores scanned barcode along with metadata like level and timestamp

### **6. Project Structure**



## 7. Key Features

- Scan Barcodes using mobile camera
- **Submit Data** to server instantly
- **Display Saved Batches** in table format
- **Delete Entries** with one click
- Persistent Storage in MongoDB

# 8. Working Process

- 1. User opens the app and enters level information.
- 2. Barcode is scanned via the camera.
- 3. Barcode and level are submitted to the backend.
- 4. Backend saves the entry in MongoDB.
- 5. User can view all scanned batches in a table format.
- 6. Option to delete any batch entry if needed.

#### 9. Sample API Endpoints

# **Method Endpoint Function**

POST /scan Save barcode + level

GET /scans Retrieve all entries

### 10. Challenges Faced

- Integrating camera and handling permissions in Expo
- Managing data flow between frontend and backend
- Handling asynchronous state updates and errors
- Designing a clean UI with FlatList and custom components

#### 11. Screens Overview

HomeScreen: UI for adding level and scanning barcodes

Camera Modal: Live camera feed using expo-camera

• BatchTable Component: Dynamic table view with delete action

#### 12. Future Enhancements

- Add user authentication (JWT)
- Export scanned data to Excel/PDF
- Cloud deployment for backend (e.g., Render or Railway)
- Offline scan support and local caching

#### 13. Conclusion

BarCodeScanApp offers a scalable and efficient solution for scanning and managing batch data through an intuitive mobile interface. With its full-stack implementation, it demonstrates practical understanding of modern development practices using React Native and Node.js.

# 14. Developer Info

Name: Sundram Pandey

**Email:** sundrampandey347@gmail.com

**GitHub:** github.com/sundram-29