# Frontend Documentation for Degree Management Platform

# **Table of Contents**

- 1. Overview
- 2. Folder Structure
- 3. Page Components and Functionalities
- 4. UI Mockups (ASCII Representations)
- 5. API Endpoints
- 6. Component Interaction Flow
- 7. State Management Strategy

## 1. Overview

The Degree Management Platform is a web-based system designed to manage student degrees efficiently. It includes user authentication, student management, degree issuance, and administrative controls. The frontend is built with React (Vite), TailwindCSS for styling, and Context API for state management.

## 2. Folder Structure

```
degree-management-platform/
- public/
                   # Public assets
I ⊢ index.html
                    # Root HTML template
# PWA manifest file
☐ I assets/
                    # Static files (logos, images)
⊢ ≡ src/
                  # Application source code
   - components/
                       # Reusable UI components
   | | common/
                        # Common UI components (Sidebar, Navbar, etc.)
     # Components specific to Dashboard
   | | | students/
                      # Components for Student Management Page
    # Components for Degree Issuance Page
    └ 📁 settings/
                      # Components for Admin/User Management
   ► pages/
                     # Page-level components
   ► = services/
                     # API calls & backend communication logic
   ► i context/
                     # Global state management (Context API)
   ⊢ 📁 utils/
                   # Helper functions (formatting, constants, etc.)
   hooks/
                     # Custom React hooks for logic reuse
| | image | routes/
                    # Routing configurations (React Router)
App.jsx
                     # Main application component
| | main.jsx
                     # Entry point
└ styles/
                  # Tailwind & global CSS configurations
```

# 3. Page Components and Functionalities

# A. Dashboard Page (Dashboard.jsx)

#### Components:

- StatsOverview.jsx: Displays student status distribution (pie chart)
- DepartmentStatsTable.jsx: Shows department-wise student performance
- DegreePreviewBox.jsx: Provides a preview of a degree certificate

#### B. Student Management Page (StudentManagement.jsx)

#### Components:

- StudentTable.jsx: Displays list of pending students
- MoveToQueueButton.jsx: Moves selected students to gueued status
- StatusIndicator.jsx: Shows status changes

## C. Degree Issuance Page (DegreeIssuance.jsx)

#### Components:

- QueuedStudentsTable.jsx: Displays students queued for degree issuance
- PreviewDegreeButton.jsx: Allows admin to preview a degree before sending
- SendDegreeButton.jsx: Sends degrees to students
- MarkAsIssuedButton.jsx: Bulk action to issue all selected degrees
- DegreeStatusTracker.jsx: Logs total students and status transitions

## D. Settings Page (Settings.jsx)

#### Components:

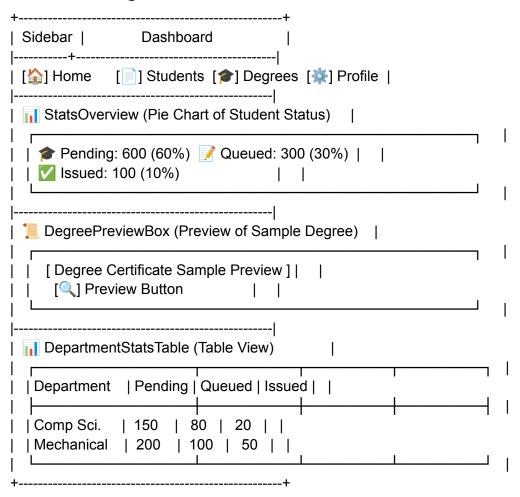
- AdminList.jsx: Lists all administrators
- AddAdminForm.jsx: Allows adding new admins
- ProfileSettings.jsx: Allows profile updates

## E. Authentication (Login.jsx)

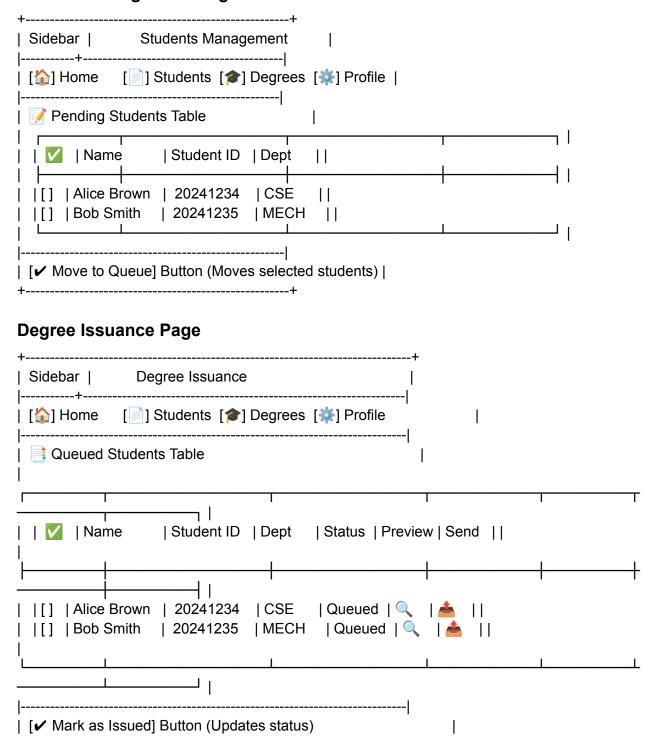
Login functionality for admins

# 4. UI Mockups (ASCII Representations)

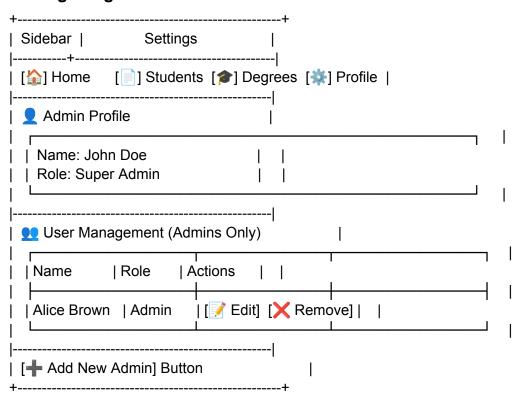
## **Dashboard Page**



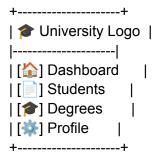
#### **Student Management Page**



## **Settings Page**



# **Sidebar (Constant in All Pages)**



# 5. API Endpoints

#### **Authentication**

- POST /api/auth/login: Admin login
- GET /api/auth/profile: Get logged-in admin profile

## **Student Management**

- GET /api/students/pending: Fetch pending students
- PATCH /api/students/move-to-queue: Move selected students to queue

## **Degree Issuance**

- GET /api/degrees/queued: Fetch queued students
- POST /api/degrees/send: Send selected degrees
- PATCH /api/degrees/mark-as-issued: Mark selected degrees as issued

## **Admin Management**

- GET /api/admins: Fetch list of admins
- POST /api/admins/add: Add a new admin

# 6. Component Interaction Flow

- Pending students → Queued (Moved via MoveToQueueButton.jsx)
- 2. Queued students → Issued (Sent via SendDegreeButton.jsx)
- 3. **Dashboard** updates in real-time to reflect student status changes

4. Admin can track status via DegreeStatusTracker.jsx

# 7. State Management Strategy

Context API is used for state management.

- AuthContext: Manages admin authentication state
- StudentContext: Handles student data fetching and updates
- **DegreeContext**: Tracks degrees' statuses and updates them dynamically