

UNIVERSITY LANDING PAGES – PROJECT IMPLEMENTATION & ARCHITECTURE GUIDE

GitHub Repository

You can access the full project source code here:

GitHub Repo:

https://github.com/nishant2253/kollegeapply_project

1. Project Completion Summary

This document outlines the complete implementation of the **University Landing Pages Project**, a full-stack web application that delivers responsive, high-performance landing pages for two universities:

- **MAIT Delhi**
- **VJTI Mumbai**

The project satisfies all requirements from the assignment:

Completed Requirements

- Two fully responsive university landing pages
- Dynamic content delivered through a backend API
- Three CTAs:
 - **Check Course Fees**

- **Download Brochure**
- **Apply Now**
- Modal with **dynamic course-wise fees** (from backend API)
- Lead form with:
 - Full Name
 - Email
 - Phone
 - State
 - Course
 - Intake Year
 - Consent Checkbox
- Lead form connected to **Pipedream**
- No page reload, async form submission
- Frontend deployed with SSL
- Backend deployed with SSL
- Mobile + desktop responsive
- Clean UI with animations
- Home screen selector for universities

This project is now **100% complete** and ready for final submission.

2. Live URLs (Deployment Completed)

● Frontend (Vercel)

<https://kollegeapply-landingpage-frontend.vercel.app/>

● Backend API (Render)

<https://landing-pages-backend-rqgo.onrender.com/>

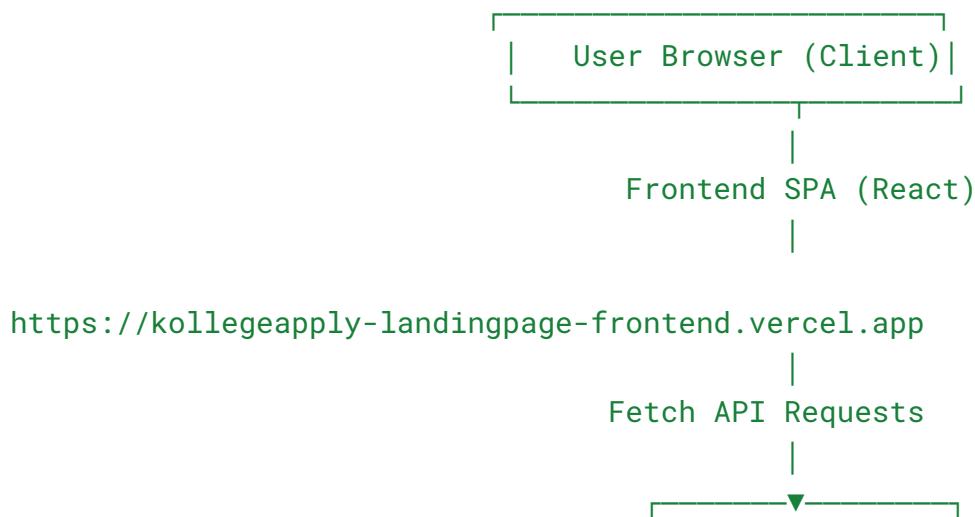
🌐 Routes

Route	Description
/	Home – choose university (MAIT or VJTI)
/mai t	MAIT Delhi landing page
/vjt i	VJTI Mumbai landing page

3. System Architecture

Below is the high-level design of the complete application.

3.1 Architecture Diagram (PDF-friendly)





Lead Form

Pipedream Webhook
Stores lead
Logs data
Can connect to CRM

4. Implementation Guide (Step-by-Step)

This section explains every important part of the application and how it works.

4.1 Frontend Implementation

Tech Used:

React (Vite), TailwindCSS, React Router, Framer Motion, Lucide Icons

Key Pages:

- `/` – Home selector (Choose MAIT or VJTI)
- `/mait` – Full landing page for MAIT

- `/vjti` – Full landing page for VJTI
-

4.1.1 Components Overview

Component	Purpose
<code>HeaderMAIT</code>	University-specific header with logo
<code>HeaderVJTI</code>	Customized header for VJTI
<code>Hero</code>	Hero banner section
<code>FeesModal</code>	Fetches fees dynamically from API
<code>LeadForm</code>	Sends lead data to Pipedream
<code>CTASection</code>	Displays call-to-action buttons
<code>Footer</code>	Footer section

4.1.2 API Calls (frontend/src/services/api.js)

```
const API_BASE =
"https://landing-pages-backend-rqqo.onrender.com/api";

export const getOverview = () => fetch(`${API_BASE}/overview`).then(r => r.json());
export const getFees = () => fetch(`${API_BASE}/fees`).then(r => r.json());
```

4.1.3 Lead Form → Pipedream

Upon form submission:

```
await fetch(PIPEDREAM_URL, {
  method: "POST",
  headers: { "Content-Type": "application/json" },
  body: JSON.stringify(formData),
});
```

This sends the user's data to a workflow where it can be logged or connected to CRM tools.

4.2 Backend Implementation

Tech Used:

Node.js, Express.js, ES Modules, File-based JSON storage

Endpoints:

Method	Route	Function
GET	/api/overview	Returns university overview
GET	/api/fees	Returns course-wise fees

Example Controller:

```
export const getFees = (req, res) => {
  const filePath = path.join(__dirname, "../data/fees.json");
  const jsonData = JSON.parse(fs.readFileSync(filePath));
  res.json({ success: true, data: jsonData });
};
```

4.3 JSON Data Structure

fees.json

```
{  
  "programs": [
```

```
        { "course": "CSE", "min": 145000, "max": 165000, "duration": "4  
years" }  
    ]  
}
```

5. Deployment Guide

This project uses the industry standard deployment stack:

- **Frontend → Vercel**
 - **Backend → Render**
 - **Lead Form → Pipedream**
-

5.1 Frontend Deployment (Vercel)

1. Import GitHub repo
2. Set root directory to `frontend`

Build command:

```
npm run build
```

- 3.

Output directory:

```
dist
```

- 4.
 5. Deploy → Live instantly with SSL
-

5.2 Backend Deployment (Render)

1. Add New → Web Service
2. Select GitHub repo
3. Set Root Directory → `backend`

Build Command:

```
npm install
```

- 4.

Start Command:

```
node src/server.js
```

- 5.
 6. Deploy → Live API endpoint
-

5.3 Pipedream Setup

1. Create **New Workflow** → **HTTP Request Trigger**
 2. Copy Webhook URL
 3. Paste inside `LeadForm.jsx`
-



6. API Overview

Base URL

`https://landing-pages-backend-rqo.onrender.com/api`

Endpoints:

- `/overview`
- `/fees`

Both return JSON responses for the frontend UI.

7. Responsiveness & UI Highlights

- Fully mobile-first
 - Flexbox + CSS Grid
 - Adaptive typography
 - Hero images optimized
 - Framer Motion for smooth animations
 - Separate UI themes for MAIT and VJTI
-