

SMART CAMPUS MANAGEMENT

# HOD Approval System

Streamlining Faculty Early Departure Requests  
with Real-Time Notifications & Digital Exit Passes

---

TEAM 3, CSE, NMIT | 2026

# PROJECT OVERVIEW

## ✗ The Problem

- Manual paper-based approval process
- HOD unavailable when faculty needs to leave early
- No record-keeping or audit trail
- Faculty unsure of approval status
- No digital exit pass or verification
- Time-consuming, error-prone workflow

## ✓ Our Solution

- Fully digital web-based workflow
- Instant email notifications to HOD & Faculty
- QR code digital exit pass generation
- Real-time status tracking dashboard
- Supporting document uploads
- Complete audit trail with timestamps

# TECHNOLOGY STACK

Modern full-stack architecture built for scale and reliability



## FRONTEND

### React + TypeScript

Vite • Tailwind CSS  
React Router • React Hot Toast



## BACKEND

### Node.js + Express

TypeScript • RESTful API  
nodemon • JWT Auth



## DATABASE

### MongoDB Atlas

Mongoose ODM  
Cloud-hosted • Atlas Free Tier



## EMAIL

### Nodemailer + Gmail

SMTP Integration  
HTML Templates • Auto Notify



## QR CODE

### qrcode Library

Dynamic Generation  
Full JSON Data Embedded



## AUTH

### JWT + bcrypt

Role-Based Access  
FACULTY • HOD • ADMIN

# KEY FEATURES



## HOD Smart Dashboard

Sort by urgency/time, filter by faculty, search, tabs for Pending / Approved / Rejected history



## Digital Exit Pass

Auto-generated QR code with full details — faculty name, date, time, reason, department



## Email Notifications

Instant Gmail alerts to HOD on new requests; Faculty gets approval/rejection email with QR code



## Document Upload

Faculty can attach PDF, DOC, JPG, PNG supporting documents up to 3 files × 5 MB each



## Duplicate Prevention

One request per date enforced — backend + frontend validation with clear error messages



## Audit Trail

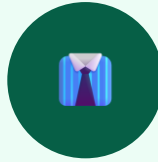
Every action logged with who, what, and when — complete request lifecycle history

# THREE USER ROLES



## FACULTY

- Submit early departure requests
- Upload supporting documents
- Track request status in real-time
- Receive email on approval/rejection
- View digital exit pass with QR code
- See HOD rejection reasons & comments



## HOD

- View prioritised pending requests
- Sort by urgency / time / faculty name
- Filter by faculty or urgency level
- Approve → auto-generate exit pass
- Reject with mandatory reason
- Request more information from faculty



## ADMIN

- Full system oversight dashboard
- Create Faculty & HOD accounts
- Manage all 10 departments
- View & delete user accounts
- Monitor all requests system-wide
- Access complete statistics

# HOW IT WORKS

End-to-end request lifecycle in 6 simple steps



# SYSTEM ARCHITECTURE

## FRONTEND TIER

- React 18 + TypeScript
- Tailwind CSS Styling
- React Router v6
- Vite Build Tool
- Deployed on Vercel

## BACKEND TIER

- Node.js + Express
- RESTful API Endpoints
- JWT Authentication
- Multer File Upload
- Nodemailer Email
- Deployed on Render

## DATA TIER

- MongoDB Atlas (Cloud)
- Mongoose ODM
- User & Request Models
- AuditLog & Comments
- Notification Schema
- Free M0 Cluster



# SECURITY & VALIDATION

Enterprise-grade security built into every layer



## JWT Authentication

- Secure token-based login
- Tokens expire automatically
- Refresh on each request



## Role-Based Access

- 3 roles: ADMIN, HOD, FACULTY
- Middleware enforces permissions
- Protected API endpoints



## Password Security

- bcrypt hashing (salt rounds 10)
- Passwords never stored plain
- Secure reset flow



## File Upload Security

- MIME type validation
- 5 MB size limit per file
- Max 3 files per request



## Input Validation

- Duplicate date prevention
- Required field enforcement
- HOD uniqueness per dept



## Data Protection

- Helmet.js HTTP headers
- CORS origin restriction
- MongoDB injection safety

# FREE DEPLOYMENT

## ▲ VERCEL — Frontend

- 1. Push frontend/ to GitHub repo
- 2. Go to vercel.com → Import Project
- 3. Set Root Directory: frontend
- 4. Build Command: npm run build
- 5. Add Environment Variable:  
VITE\_API\_URL = https://your-app.onrender.com/api
- 6. Click Deploy — Done! ✓

## 🏠 RENDER — Backend

- 1. Push backend/ to GitHub repo
- 2. Go to render.com → New Web Service
- 3. Connect GitHub → select repo
- 4. Root Directory: backend
- 5. Build: npm install && npm run build
- 6. Start: node dist/server.js
- 7. Add all .env variables → Deploy ✓

# Thank You

## HOD APPROVAL SYSTEM

3

User Roles

10

Departments

15+

API Endpoints

100%

Free Hosting

- ✓ Complete Request Lifecycle Management
- ✓ Real-Time Email Notifications (Gmail SMTP)
- ✓ QR Code Digital Exit Passes
- ✓ Supporting Document Uploads
- ✓ Deployed FREE on Vercel + Render + MongoDB Atlas