

# UWC PostGrad Portal

## README — Project Documentation

*University of the Western Cape*

Generated: 8 February 2026

Project: Postgraduate Request Portal

Methodology: Design Science Research (DSRM)

# Table of Contents

---

<b>UWC PostGrad Portal</b>	3
Features	3
Four User Roles	4
Core Functionality	4
Tech Stack	4
<b>Install dependencies</b>	4
<b>Configure environment variables</b>	4
<b>Edit .env with your Firebase and EmailJS credentials (see below)</b>	4
<b>Start development server</b>	4
<b>Production build</b>	5
<b>Preview production build</b>	5
Environment Variables	5
Demo Accounts	5
Project Structure	6
Routes	8

# UWC PostGrad Portal

```
<p align="center">

</p>
```

A role-based postgraduate request management system for the University of the Western Cape (UWC). Built with React 19, Vite, Firebase (Authentication + Cloud Firestore), and standard CSS.

## Features

Role	Capabilities
Student	Create/submit HD requests, track submissions, manage milestones, view academic progress
Supervisor	Review requests, approve/refer back, sign digitally, nudge students, annotate documents
Coordinator	Manage all requests, forward to Faculty/Senate Boards, record outcomes, export agendas
Admin	System overview, analytics, role management, audit logs, dataset exports

- \* HD Request Workflow -- Full lifecycle: draft -> supervisor review -> co-supervisor sign -> coordinator -> Faculty Board -> Senate Board -> approved
- \* Access Code System -- Secure supervisor access via generated 6-character codes with expiry
- \* Digital Signatures -- Draw or type signature pad for approval actions
- \* Submission Tracker -- Visual workflow progress bar with owner tracking and response timers
- \* Document Review & Version Control -- Multi-version document management with visual diff, feedback, and status workflow (submitted -> under review -> changes requested -> approved)
- \* PDF Annotation System -- Full-screen PDF viewer (react-pdf) with text selection, inline highlight annotations, colour picker, reply threads, resolve/reopen, and draft -> sent batch workflow
- \* Batch Annotation Workflow -- Supervisors/coordinators save annotations as drafts, review all at once, then confirm & send to notify the student via in-app notification + email
- \* Nudge System -- Supervisors can send reminder notifications to students
- \* Committee Exports -- CSV export of Faculty/Senate Board agendas with student number, degree, and supervisor
- \* Refer-Back Workflow -- 24-hour amendment timer when requests are referred back
- \* Calendar -- Full CRUD calendar with month view, role-based auto-filtering, and event types
- \* Milestones -- Students can log academic milestones (conferences, publications, etc.)

- 
- \* Audit Logs -- Searchable activity log with date filtering and CSV export
  - \* Analytics Dashboard -- Bar charts for request status/type distribution, summary statistics
  - \* Role Management -- Admin interface to reassign user roles
  - \* Overdue Monitoring -- "Overdue Only" filter on requests page for coordinators/admins
  - \* Notifications -- Real-time Firestore-backed notifications with bell icon, unread count, mark-read, and link navigation
  - \* Email Notifications -- EmailJS integration for real email delivery on all cross-user actions (submissions, approvals, annotations, comments, feedback, nudges)
  - \* Dark/Light Mode -- Complete theming system with persistent preference via Settings -> Appearance tab. 100+ element-level CSS overrides across all pages ensure consistent styling in both themes
  - \* Help & Documentation -- Dedicated Help & Docs page (all roles) with 7 FAQ categories (20+ questions), 6 static guides, and 13 interactive walkthrough tours
  - \* Guided Tour System -- Overlay-based walkthrough engine with SVG mask highlighting, auto-scroll to elements, click-to-proceed steps, cross-page navigation, and 4 role-specific full system tours

## Four User Roles

### Core Functionality

### Tech Stack

## Install dependencies

```
npm install
```

## Configure environment variables

```
cp .env.example .env
```

## Edit .env with your Firebase and EmailJS credentials (see below)

## Start development server

---

```
npm run dev
```

## Production build

```
npm run build
```

## Preview production build

```
npm run preview
```

Create a .env file in the project root:

```
VITE_FIREBASE_API_KEY=your-api-key  
VITE_FIREBASE_AUTH_DOMAIN=your-project.firebaseio.com  
VITE_FIREBASE_PROJECT_ID=your-project-id  
VITE_FIREBASE_STORAGE_BUCKET=your-project.firebaseiostorage.app  
VITE_FIREBASE_MESSAGING_SENDER_ID=your-sender-id  
VITE_FIREBASE_APP_ID=your-app-id  
  
VITE_EMAILJS_SERVICE_ID=your-emailjs-service-id  
VITE_EMAILJS_TEMPLATE_ID=your-emailjs-template-id  
VITE_EMAILJS_PUBLIC_KEY=your-emailjs-public-key
```

Email	Role	Name
student@uwc.ac.za	Student	Thabo Molefe
student2@uwc.ac.za	Student	Amahle Dlamini
student3@uwc.ac.za	Student	Sipho Dlamini
supervisor@uwc.ac.za	Supervisor	Prof. Sarah van der Berg
supervisor2@uwc.ac.za	Supervisor	Dr. Marcus Thompson
coordinator@uwc.ac.za	Coordinator	Dr. Fatima Patel
admin@uwc.ac.za	Admin	Linda Mkhize

*Default password: Portal@2026*

## Environment Variables

### Demo Accounts

---

## Project Structure

```
postgrad-portal/
--- index.html # Entry HTML (UWC favicon)
--- package.json
--- firebase.json # Firebase project config
--- firestore.rules # Firestore security rules
--- vite.config.js
--- public/
- --- uwc_logo.svg # University of the Western Cape logo
- --- documents/ # 19 generated sample documents (PDF/DOCX/XLSX)
- --- hdr-001/ # Progress_Report, Publication_Evidence, Supervisor_Feedback
- --- hdr-002/ # Research_Proposal, Literature_Review, Ethics_Clearance
- --- hdr-003/ # Extension_Motivation_Letter
- --- hdr-004/ # Ethics_Application, Informed_Consent, Data_Collection
- --- hdr-005/ # Registration, Academic_Transcript, Progress_Report
- --- hdr-006/ # Examiner_Nomination, Turnitin_Report
- --- hdr-007/ # Progress_Report_NK
- --- hdr-009/ # Medical_Certificate
- --- hdr-011/ # Title_Registration_Form
- --- hdr-012/ # Progress_Report_Jan2026
--- scripts/ # Automation & provisioning
- --- setup-firebase.mjs # Firebase Auth user creation
- --- seed-firebase.mjs # Firestore collection seeding (7 collections)
- --- reseed-firebase.mjs # Clear & re-seed all data
- --- seed-document-versions.mjs # Document version history seeding
- --- seed-annotations.mjs # Annotation seed data
- --- upload-documents.mjs # Generate 13 sample PDFs/DOCX/XLSX with pdf-lib
- --- generate-missing-pdfs.mjs # Generate 6 additional PDFs referenced by versions
- --- test-emailjs.mjs # EmailJS connectivity test
- --- test-firebase.mjs # Firestore connectivity test
--- docs/
- --- FIREBASE_MIGRATION_CHANGELOG.md
- --- EMAILJS_SETUP.md
--- src/
--- App.jsx # Routes and protected route wrapper
--- main.jsx # Entry point (AuthProvider + DataProvider)
--- index.css # Global styles & CSS custom properties
--- components/
- --- common/ # Shared UI - Card, Modal, StatusBadge, Avatar, etc.
- - --- index.jsx
- - --- common.css
- - --- SignaturePad.jsx
```

---

```
- --- layout/ # App shell - sidebar, header, outlet
- ---- Layout.jsx
- ---- Header.jsx # Notification bell dropdown
- ---- Sidebar.jsx # UWC-branded nav with role-specific menus
- ---- layout.css
- --- AnnotatedDocViewer.jsx # Full-screen PDF viewer + annotation system
- --- AnnotatedDocViewer.css
--- context/
- --- AuthContext.jsx # Firebase Auth provider
- --- DataContext.jsx # Firestore real-time subscriptions + mutations
- --- ThemeContext.jsx # Dark/light mode provider (localStorage persistence)
- --- GuidedTour.jsx # Guided tour engine (overlay, highlight, auto-scroll)
- --- GuidedTour.css # Tour overlay + tooltip styles
--- firebase/
- --- config.js # Firebase app initialisation
- --- firestore.js # Firestore CRUD (all collections)
- --- collections.js # Collection name constants
- --- documentVersions.js # Document version control operations
- --- annotations.js # Annotation CRUD + batch confirm/send
- --- storage.js # Firebase Storage helpers
--- services/
- --- emailService.js # EmailJS integration (send real emails)
- --- pdfService.js # Client-side PDF generation (jsPDF)
--- pages/
- --- Dashboard.jsx # Role-based dashboard router
- --- HDRequestsPage.jsx # Request list, detail modal, all workflow actions
- --- DocumentReviewPage.jsx # Version control, comments, feedback, annotations
- --- SubmissionTracker.jsx # Visual workflow progress tracker
- --- CalendarPage.jsx # Month calendar with CRUD
- --- StudentsPage.jsx # Student directory with edit modal
- --- AcademicProgressPage.jsx # Student academic history
- --- AnalyticsPage.jsx # Admin analytics with charts
- --- AuditLogsPage.jsx # Searchable audit log
- --- RoleManagementPage.jsx # Admin role assignment
- --- LoginPage.jsx # Login with demo quick-access
- --- SettingsPage.jsx # Profile, notifications, password, appearance (theme)
- --- HelpPage.jsx # Help & Docs (FAQs, guides, interactive walkthroughs)
- --- HelpPage.css # Help page styles
- --- walkthroughs.js # 13 walkthrough definitions (4 full + 9 task-specific)
- --- SeedPage.jsx # Admin-only database reseed tool
--- dashboards/
- --- StudentDashboard.jsx
- --- SupervisorDashboard.jsx
- --- CoordinatorDashboard.jsx
```

---

```
- --- AdminDashboard.jsx
--- utils/
--- constants.js # Status configs, labels, workflow states
--- helpers.js # Date formatting, utilities
```

## Routes