Internship at Centre for Innovation & Entrepreneurship, PES University

"Development, Deployment and Maintenance of the MedInnTech Programme Website"

Submitted by

Rahul Biju B.

SRN: PES2UG24CS393

Under the guidance of

Prof. Lakshmeesha

Professor and Director

Centre for Innovation & Entrepreneurship

PES University – Electronic City Campus

8 June 2026 – 3 July 2026 (4 Weeks)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City Campus, Bengaluru, Karnataka, India

COMPANY CERTIFICATE

The official Company Internship Completion Certificate
(or the first page of the Joining/Acceptance Letter, if permitted)
is to be attached here by the student before final submission.

DECLARATION

I hereby declare that the project entitled “Development, Deployment and Maintenance of the MedInnTech Programme Website” was carried out at the Centre for Innovation & Entrepreneurship, PES University, Electronic City Campus, by me under the guidance of Prof. Lakshmeesha, Professor and Director, Centre for Innovation & Entrepreneurship, during the internship period from 8 June 2026 to 3 July 2026.

The work presented in this report records my responsibilities and learning as the primary contributor to the project repository. It includes the design and implementation of the website, repeated changes made in response to client feedback, responsive-layout engineering, deployment, maintenance and related coordination activities.

This report is submitted in partial fulfilment of the internship requirements for the Bachelor of Technology programme in Computer Science and Engineering at PES University, Bengaluru. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree or diploma.

Place: Bengaluru

Signature of the Student:

_____

Date: _____27/7/26_____

Rahul B

SRN: PES2UG24CS393

ACKNOWLEDGEMENT

I express my sincere gratitude to Prof. Lakshmeesha, Professor and Director, Centre for Innovation & Entrepreneurship, PES University – Electronic City Campus, for entrusting me with the MedInnTech website project and for his continuous guidance, technical discussions, feedback and encouragement. His direct involvement, including extended office sessions during critical stages of the project, helped me convert evolving requirements into a stable deployed website.

I am grateful to the Internship Coordinators, Dr. Animesh Giri and Dr. Prema R., Department of Computer Science and Engineering, PES University – Electronic City Campus, for organizing and supporting the internship process.

I thank my Capstone Project Guide, [Name of Capstone Project Guide], Department of Computer Science and Engineering, PES University – Electronic City Campus, for academic guidance and encouragement. This placeholder may be updated before submission.

I extend my gratitude to Dr. Sandesh B. J., Chairperson, Department of Computer Science and Engineering, PES University – Electronic City Campus, for the support and learning opportunities provided by the department.

I thank Dr. B. K. Keshavan, Dean of Faculty, PES University, for the institutional support provided to students.

I am deeply grateful to Prof. Jawahar Doreswamy, Chancellor, PES University; Dr. Suryaprasad J., Vice Chancellor, PES University; and Prof. Nagarjuna Sadineni, Pro-Vice Chancellor, PES University, for creating an environment that encourages practical learning, innovation and industry exposure.

Finally, I thank my parents and friends for their patience, motivation and support throughout the internship, especially during days that required extended working hours and weekend coordination.

ABSTRACT

This report presents the work completed during a four-week Full-Stack Development internship at the Centre for Innovation & Entrepreneurship (CIE), PES University – Electronic City Campus. The principal assignment was to develop the public-facing website for the Medical Innovations and Technologies (MedInnTech) initiative from the ground up and to carry it through iterative client review, deployment and maintenance.

The project was implemented as a React single-page application built with Vite, React Router, Tailwind CSS and a component-oriented user-interface architecture. The final application contains a broad set of public routes for programme information, course pathways, curriculum, faculty, media, industry engagement, frequently asked questions, announcements, downloads, registration and cohort-specific information. Across the development lifecycle, more than twenty webpages and page variants were produced as requirements changed. The final router consolidates the delivery into nineteen active public route entries, supported by shared header, footer, page-shell, loading and announcement components [1].

A major engineering challenge was ensuring that content-heavy layouts retained the client-approved composition on laptops, desktops, tablets and mobile devices. The solution combined CSS clamp() functions, viewport and container-relative units, adaptive grids, height-aware media queries, ResizeObserver-based measurement and reference-canvas scaling. These techniques were used to reduce clipping, inconsistent reflow and sudden typography jumps while retaining visual consistency.

As the primary contributor, I recorded 167 commits and a contribution summary of 50,305 additions and 38,944 deletions. I also managed build releases, deployment follow-up and maintenance fixes; worked with my mentor until approximately 7:00 p.m. on selected days; and remained available on calls during two weekends to address time-sensitive requirements. The internship developed my abilities in modern web engineering, requirement interpretation, client communication, responsive design, Git-based iteration, deployment ownership and professional follow-through.

TABLE OF CONTENTS

LIST OF TABLES, FIGURES AND ABBREVIATIONS

List of Tables

List of Figures

Abbreviations

Abbreviation	Expansion
CIE	Centre for Innovation & Entrepreneurship
SPA	Single-Page Application
UI	User Interface
UX	User Experience
SEO	Search Engine Optimization
CSS	Cascading Style Sheets
RWD	Responsive Web Design
PDF	Portable Document Format
CGPA	Cumulative Grade Point Average
API	Application Programming Interface

CHAPTER 1: INTRODUCTION AND ORGANIZATION PROFILE

1.1 Introduction

An internship provides a bridge between academic study and the constraints of a live engineering assignment. My internship at the Centre for Innovation & Entrepreneurship, PES University – Electronic City Campus, focused on the end-to-end delivery of a public website for the MedInnTech initiative. The assignment was not limited to preparing isolated screens. It required translating frequently changing client expectations into maintainable code, coordinating with a mentor, handling the full release cycle and supporting the deployed product after changes.

1.2 Organization Profile

The Centre for Innovation & Entrepreneurship is a university-based innovation unit that supports practical development, interdisciplinary projects and entrepreneurship-oriented initiatives. It operates within PES University and connects academic expertise with programme development and real implementation needs. The internship team was compact and mentor-led, which gave me direct access to the project decision-maker and made each iteration closely tied to client feedback. The centre's sector can be described as higher education, innovation support and technology development.

The project supported Medical Innovations and Technologies (MedInnTech), an interdisciplinary initiative positioned at the intersection of engineering, healthcare, artificial intelligence, medical devices, biomedical imaging and digital health. The website had to communicate programme pathways, curriculum, cohort information, industry relationships, contact details and application steps to students, professionals and other stakeholders.

1.3 Internship Objectives

- Develop a complete, professional and responsive website from the ground up.
- Translate changing client specifications into working page designs without losing consistency.
- Create reusable components and a maintainable route structure for a large multi-page experience.
- Deploy the application, monitor its behaviour and make maintenance corrections.
- Gain practical experience in full-stack web-development workflow, communication and release ownership.

The internship lasted four weeks, from 8 June 2026 to 3 July 2026. I worked under the direct guidance of Prof. Lakshmeesha and served as the primary contributor to the repository and delivered application.

CHAPTER 2: INTERNSHIP CONTEXT, ROLE AND RESPONSIBILITIES

2.1 Internship Particulars

Particular	Details
Intern	Rahul Biju B. (PES2UG24CS393)
Position	Summer Intern – Full-Stack Development
Organization	Centre for Innovation & Entrepreneurship, PES University
Mentor	Prof. Lakshmeesha
Duration	4 weeks: 8 June 2026 to 3 July 2026
Mode	Hybrid, with substantial office-based collaboration
Primary project	MedInnTech programme website
Repository role	Primary contributor and release maintainer

2.2 Role within the Team

My role combined implementation ownership with day-to-day requirement handling. Because I was the primary contributor, most feature decisions had to be converted into code by me, tested, shown to the mentor, revised and then released. I was responsible for the application structure, page implementation, visual consistency, responsive behaviour, integration of assets and documents, route-level metadata, deployment follow-up and maintenance.

2.3 Main Responsibilities

- Create the website from an empty project base and establish the React/Vite structure.
- Develop more than twenty webpages and page variants during the delivery cycle according to client requirements.
- Build shared navigation, branding, page-shell, footer, loading and announcement components.
- Implement course, cohort, curriculum, faculty, media, industry, FAQ, downloads, contact and registration experiences.
- Resolve layout, clamping, resizing and viewport-height issues across different screen sizes.
- Maintain Git history, prepare production builds, manage deployment and support post-deployment changes.
- Communicate progress and clarifications with the mentor, including selected late-evening office sessions and weekend calls.

CHAPTER 3: PROJECT OVERVIEW, REQUIREMENTS AND SCOPE

3.1 Project Need

The MedInnTech initiative required a central website that could explain multiple programmes and serve different audiences without appearing fragmented. The client needed a modern PES-branded interface that could present detailed academic content while remaining usable on laptops, desktops and mobile phones. Requirements changed repeatedly as programme details, cohort dates, course offerings, images, fee information and layout expectations were refined.

3.2 Requirement Handling

The development process was iterative rather than linear. New client inputs sometimes required a local text edit, while others changed the entire page composition. I therefore maintained a reusable component base but allowed selected pages—especially the Minor Degree, second-cohort and Technical Executive pages—to use specialised layouts. This balance reduced unnecessary duplication while still permitting highly specific client-approved compositions.

3.3 Functional Scope

Route / Module Group	Purpose
Programme discovery	Landing page, programme overview, about and course comparison
Course pathways	Minor Degree, second cohort, Masters, Executive and Technical Executive pages
Academic information	Curriculum, schedules, fee information and downloadable resources
Engagement	Industry, industry connect, alumni, faculty, media and announcements
Support and conversion	FAQ filtering, contact details, registration redirect and document links
Shared behaviour	Responsive navigation, page loader, metadata, reveal animation and common footer

3.4 Scope Boundaries

The final repository is primarily a client-side React application. It integrates external services such as Google Forms and static PDF documents, but it does not contain a custom application server or database. Accordingly, the “full-stack” learning in this internship covered the complete delivery workflow—UI development, routing, integration, production build, deployment and operational maintenance—rather than the construction of a separate backend service. This distinction is important for accurately describing the delivered system.

CHAPTER 4: SYSTEM DESIGN AND TECHNOLOGY STACK

4.1 Architectural Design

The website follows a modular SPA architecture. The entry point mounts the React application, while the application router lazy-loads page modules and displays a skeleton loader during transitions. Shared components provide navigation, branding and page structure. Individual route files hold page-specific content and interactions, while static assets and PDF schedules are bundled by Vite. A registration route redirects users to an external Google Form [1].

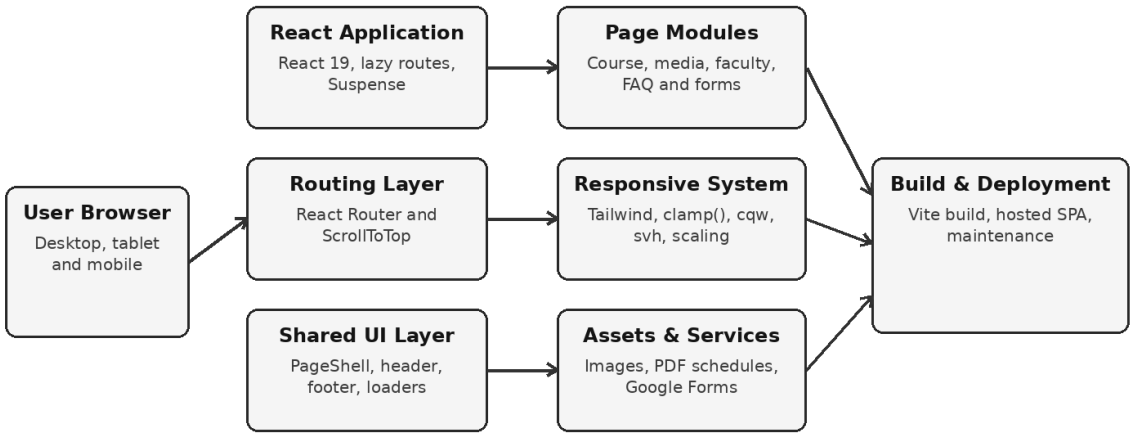

Figure 1: Application architecture and deployment flow

4.2 Technology Stack

Technology	Use in Project
React 19	Component rendering, state and effects
React Router 6	Client-side routing and nested course routes
Vite 7	Development server and production build
Tailwind CSS 4	Utility-first styling and responsive layout
Radix UI	Accessible accordion primitives
Lucide React	Consistent iconography
Sonner	Lightweight user notifications
TanStack Query	Application-level query client foundation
Git and GitHub	Version control, contribution tracking and collaboration

CHAPTER 5: IMPLEMENTATION DETAILS

5.1 Application Foundation

I created the application foundation using Vite and React. The root application configures a QueryClientProvider, BrowserRouter, route-change scroll handling and Suspense-based lazy loading. Lazy imports reduce the initial route bundle and allow the skeleton PageLoader to communicate progress during page transitions. Each route updates the document title and description metadata, improving clarity when pages are opened or shared [1].

5.2 Shared Site Components

The SiteHeader implements a two-tier PES-branded desktop header and a compact mobile menu. It combines internal NavLink navigation with external PES links, responsive logo sizing and active-route styling. The SiteFooter provides both full and compact variants, allowing content-heavy executive pages to preserve viewport space. PageShell wraps standard pages with the header, optional hero banner, content area and footer. BrandText consistently colourises the MedInnTech name, while AnnouncementsSection centralises course notices and “new” status behaviour.

5.3 Page-Level Development

Module	Implementation Highlights
Landing and overview	Hero typing effect, MedTech pillars, programme value and industry context
Course overview	Three-column, viewport-aware course comparison and learning journey
Minor Degree	Cohort cards, course overview artwork, eligibility and FAQ accordions
Second Cohort	Interactive images, PDF term schedules, dates, fees and registration calls to action
Technical Executive	Dense dashboard-style layout with dynamic available-height measurement
Media	Responsive photographic grid with glassmorphism labels
FAQ and announcements	Category filtering, two-column accordions, expandable announcement list
Registration	Immediate external-form redirect with accessible fallback link

Over the complete iteration cycle, more than twenty webpages and page variants were built. The final application consolidates the approved material into nineteen active public routes, reducing duplication while retaining all required user journeys.

5.4 Content, Assets and Document Integration

I integrated images, course graphics and cohort-specific PDF schedules directly into route modules so Vite could resolve them during production builds. The second-cohort page links four term schedules and expandable programme, course and fee graphics, while the media page uses object-fit images and gradient scrims for readable labels.

5.5 Interaction and Accessibility Considerations

Interactive features include practical accessibility fallbacks: the registration redirect retains a manual link, images use alternative text, the mobile-menu button is labelled, loading states expose status text, reduced-motion preferences disable shimmer animation, and FAQ accordions use Radix primitives.

5.6 SEO and Navigation

Each major route sets a page-specific title and meta description. React Router supplies nested course paths and active navigation, and ScrollToTop resets the viewport after route changes. The final route set covers programme, course, curriculum, faculty, media, industry, support and registration journeys [1].

5.7 Version-Control Practice

Git was used continuously as a checkpoint system for pages, content, responsive fixes, assets and releases. My record contains 167 commits, 50,305 additions and 38,944 deletions. The high deletion count reflects redesign and refactoring as the client refined the approved output.

5.8 Quality and Maintainability

Global PES colour tokens, typography utilities and shared buttons reduce style divergence. Route content is separated from structural components, and announcements are centralised. Future work should move more repeated content into structured data and add automated unit, accessibility and end-to-end tests.

CHAPTER 6: RESPONSIVE DESIGN AND CROSS-SCREEN CONSISTENCY

6.1 Nature of the Challenge

The hardest problem was preserving dense, client-approved compositions across common laptop and desktop sizes. Breakpoint-only layouts produced typography jumps, excessive page height, image misalignment and clipping under browser zoom or operating-system scaling.

6.2 Engineering Approach

The solution combined clamp() for fluid typography and spacing, cqw units for component-relative headings, svh calculations for usable height, height-aware media queries for short laptops, and flexible grid/flex reflow for mobile screens.

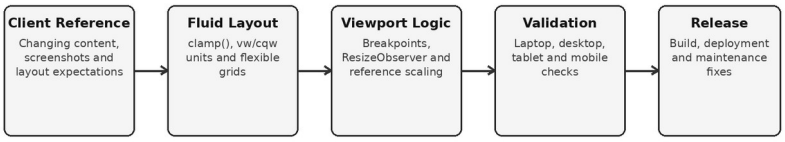

Figure 2: Responsive-layout engineering workflow

6.3 Reference-Canvas Scaling

For the second-cohort page, I used a reference width and height, measured available space after the header, and applied the smaller width/height scale to the completed layout. The Technical Executive page measures space between header and footer with ResizeObserver and recalculates after fonts load. This prevented independent card reflow and improved agreement with the design references [1].

6.4 Validation and Outcomes

I checked desktops, common laptops, tablets and mobile layouts, correcting overflow, clipping and image sizing. Fluid scaling is used by default, with measured layouts reserved for pages whose approved composition required stricter control.

CHAPTER 7: DEPLOYMENT, MAINTENANCE AND PROFESSIONAL WORK PRACTICES

7.1 Build and Deployment Ownership

My responsibility continued beyond local development. I prepared production builds, handled deployment updates and monitored the deployed website after changes. Static assets, route paths and external links had to function in the hosted environment, not only in the development server. When issues were discovered after a release, I traced them to the relevant route or responsive rule, applied a focused correction, rebuilt the application and verified the result.

7.2 Maintenance

Maintenance work included changing programme content, replacing images and PDFs, adjusting dates and fees, correcting navigation, handling new client requests and preserving consistency after each revision. Because the project was client-driven, maintenance was continuous during the internship rather than a separate final phase. This taught me to treat deployed software as an evolving product and to minimise regressions while making urgent updates.

7.3 Time Commitment and Mentor Collaboration

Several delivery points required additional coordination. On selected days, I remained in the office with my mentor until approximately 7:00 p.m. to complete changes, review the displayed result and resolve issues while feedback was immediately available on some days. I also stayed available on calls during two weekends when project decisions or time-sensitive corrections had to be completed. These experiences strengthened my ability to communicate progress honestly, prioritise critical fixes and remain dependable under deadline pressure.

7.4 Working with Changing Requirements

Frequent requirement changes were initially challenging because completed layouts could become obsolete. I learned to separate stable design rules from changeable content, ask for the most important acceptance criteria, commit working checkpoints and avoid over-investing in details that had not yet been approved. I also learned that client satisfaction depends on follow-through: an implementation is not complete merely because the code compiles; it must match the requested output, work on the target screens and remain stable after deployment.

7.5 Confidentiality and Conduct

I followed the organization’s professional expectations, maintained regular mentor communication and limited this report to information visible in the public project repository or explicitly permitted for the internship record. No private credentials, deployment secrets or confidential internal discussions are included.

CHAPTER 8: RESULTS, ACCOMPLISHMENTS AND EVIDENCE

8.1 Major Accomplishments

- Built the MedInnTech website from the ground up and carried it through repeated client-led redesigns.
- Delivered more than twenty webpages and page variants over the development cycle, consolidated into nineteen active final routes.
- Created responsive, reusable and specialised page systems for both standard informational pages and dense dashboard-style programme pages.
- Managed production builds, deployment follow-up and maintenance updates.
- Resolved difficult clamping, viewport-height and resizing problems so content remained visually consistent across screen sizes.
- Maintained a high-volume Git contribution record while serving as the primary contributor.

8.2 Quantitative and Documentary Evidence

Evidence Item	Recorded Result
Commits authored / recorded	167
Lines added	50,305
Lines removed	38,944
Active public route entries in final router	19
Webpages and page variants across lifecycle	More than 20
Internship duration	4 weeks
Weekend support periods	2 weekends
Extended office collaboration	Until approximately 7:00 p.m. on selected days

8.3 Contribution to Organizational Goals

The website gave the MedInnTech initiative a structured digital presence for communicating programme value, course options, curriculum, cohort information, industry connections and registration pathways. Reusable components and centralised styling made it possible to respond more quickly when details changed. Deployment and maintenance ownership reduced the delay between approved feedback and public availability.

8.4 Challenges Overcome

The key obstacles were evolving specifications, dense content, inconsistent behaviour across viewport dimensions and the need to preserve a reference composition. I overcame these through incremental commits, reusable abstractions, targeted measured layouts, continuous visual validation and direct mentor review. The large number of additions and deletions reflects the amount of redesign and refactoring necessary to reach the approved state.

CHAPTER 9: LESSONS LEARNED AND AREAS FOR IMPROVEMENT

9.1 Technical and Professional Skills Developed

Skill	How It Will Be Applied
Full-stack delivery workflow	Apply frontend engineering, integrations, production builds, deployment and maintenance as one connected process.
React component design	Separate shared structure from page-specific layouts and manage state/effects responsibly.
Responsive CSS	Use clamp(), viewport units, container units, measured layouts and adaptive grids.
Requirement interpretation	Convert client language and visual references into testable acceptance criteria.
Git discipline	Use commits as safe checkpoints during rapid iteration and refactoring.
Client and mentor communication	Clarify priorities, demonstrate progress and follow through on feedback.
Professional reliability	Support critical releases through extended hours and scheduled weekend coordination.
Debugging and maintenance	Trace visual and deployment defects to the smallest practical change.

9.2 Key Lessons

The most important lesson was that technical correctness and client correctness are different. A layout may be valid, reusable and responsive yet still fail if it does not match the client's intended hierarchy or composition. I learned to validate both dimensions. I also learned that responsiveness is a system-level property: typography, spacing, image behaviour, viewport height, browser scaling and navigation all interact.

9.3 Areas for Further Development

I intend to improve automated testing, accessibility auditing, performance measurement and content modelling. The project would benefit from unit tests for shared utilities, route-level end-to-end tests, automated responsive screenshot comparisons and a small content-management layer for frequently changing programme data. I also plan to strengthen backend development skills so future "full-stack" projects include secure server APIs, persistent data, authentication and observability in addition to deployment operations.

9.4 Self-Evaluation

I believe my strongest contribution was ownership. I did not stop after building initial pages; I continued through redesigns, late-stage layout problems, deployment and maintenance. My main improvement area is to establish acceptance criteria and automated checks earlier, which would reduce the cost of repeated manual validation when requirements change.

CHAPTER 10: CONCLUSION AND RECOMMENDATIONS

10.1 Conclusion

The internship at the Centre for Innovation & Entrepreneurship was a practical exercise in building and owning a real client-facing product. I developed the MedInnTech website from the ground up, implemented a large set of page experiences, adapted the work repeatedly as specifications changed and remained responsible for deployment and maintenance. The project strengthened my technical knowledge of React, Vite, routing, component design and advanced responsive CSS, while also improving my communication, time management and ability to follow client requirements through to completion.

The most valuable outcome was learning to balance reuse with precision. Shared components improved maintainability, but specialised pages required carefully measured layouts to preserve the intended design. The resulting repository demonstrates both approaches. The internship also showed me that professional software delivery requires availability, accountability and post-release support, not only coding speed.

I express my gratitude to CIE and Prof. Lakshmeesha for the opportunity and guidance. Primary professional contact: lakshmeesha@pes.edu.

REFERENCES / BIBLIOGRAPHY

[1] 2n12bq24hip1ke3d, “MedTech Website Repository (1ismtwod),” GitHub, 2026. Available: https://github.com/2n12bq24hip1ke3d/1ismtwod.

[2] Meta Platforms, Inc., “React Documentation,” https://react.dev/.

[3] Vite Contributors, “Vite Guide,” https://vite.dev/guide/.

[4] Remix Software, Inc., “React Router Documentation,” https://reactrouter.com/.

[5] Tailwind Labs, “Tailwind CSS Documentation,” https://tailwindcss.com/docs.

[6] Mozilla Developer Network, “CSS clamp(),” https://developer.mozilla.org/docs/Web/CSS/clamp.

[7] Mozilla Developer Network, “ResizeObserver,” https://developer.mozilla.org/docs/Web/API/ResizeObserver.

[8] Mozilla Developer Network, “IntersectionObserver,” https://developer.mozilla.org/docs/Web/API/IntersectionObserver.

[9] Radix UI, “Accordion Primitive,” https://www.radix-ui.com/primitives/docs/components/accordion.