Q1. Project Requirements and Intended Outcomes

When I started this project, the FGS NZ website was in rough shape broken databases, corrupted files, and missing media everywhere. Honestly, it felt like walking into a digital earthquake zone. My main goal was to bring it back to life: rebuild it using WordPress and PHP, make it scalable and maintainable, and align it with FGS branding.

But this wasn't just about "fixing" things. I had to dig into old backups, recover media files via FTP and Media Sync, and rebuild key pages like Home, About, Contact, Temple Info, and Events using Gutenberg blocks. I also needed to make sure navigation worked smoothly, layouts matched the Figma designs, and CSS was modular so future updates wouldn't turn into a headache.

By the end, my goal was to deliver a site that wasn't just functional it had to be clean, professional, and easy for staff and future developers to manage. It needed to represent FGS online in a way that felt polished, welcoming, and accessible to everyone.

Q2. Applying Industry Best Practices

I leaned on industry-standard practices throughout the project it helped me feel like I wasn't just winging it, especially when restoring something so fragile. For project management, I used Notion to track tasks by weeks and technical focus. Each task included goals, blockers, notes on shortcodes, FTP restoration, or CSS tweaks. This way, even though I wasn't on a full Agile team, I could treat each week like a mini-sprint and see steady progress.

Styling was another big focus. I set up modular CSS, so each page or component had its own scoped file. That made debugging or future edits way easier. Components like the event cards or gallery carousel could be updated independently without breaking anything else.

I also leaned on prototyping and testing. Figma was my visual guide, and WordPress previews let me check layouts, responsiveness, and accessibility before pushing pages live. For the backend, I created a custom post type for activities using CPT UI and ACF. That gave staff control over content without relying on bloated plugins, keeping things fast and manageable.

Restoring media was its own adventure. I carefully retrieved files from old backups and ensured nothing was missing or misrepresented. In the end, combining structured tracking, modular CSS, and iterative testing let me meet the project requirements efficiently while following professional development practices.

Q3. Design Thinking in Action

Even though a lot of this project was technical, I tried to keep the user experience front and centre. I found myself constantly thinking: *"How will someone actually use this site?"*

When empathising, I considered both the visitors and the staff managing content most importantly. Navigation needed to be simple, anchor links clear, and content easy to edit with Gutenberg blocks. I wanted future staff to feel confident updating pages without feeling lost in code.

For prototyping, Figma was my reference point. I translated its design tokens into block structures, built custom components like event cards and gallery carousels, and tested early in WordPress. Iterative previews helped me catch things before they became bigger issues.

I'm proud not just of the site I restored, but of the resilience it took to get here. Ten weeks of obstacles, growth, and mahi and I showed up for every single one.

Q4 & Q5. Agile Mindset and Practices

Even working solo, I found myself naturally applying Agile principles. I focused on people, staff who'd manage content, and visitors who needed an easy to use site. I stayed value-driven, prioritizing the pages and components that would have the most impact, like the Home page, About page, and key events features.

My workflow was iterative. Tasks in Notion were broken down by page or component, built incrementally, tested, and refined. Feedback from mentors or internal QA was folded in real-time. It was flexible and adaptive if I hit a blocker, I could pivot priorities without losing momentum.

This Agile approach meant that progress was steady, issues were spotted early, and the final delivery was functional, polished, and aligned with user needs. Even under time pressure, it felt manageable because I was always focusing on delivering tangible value each step of the way.

Q6. Stakeholders

A few key groups shaped this project. FGS leadership needed a stable, branded website to represent the organization publicly. Staff and administrators needed a backend that was simple and maintainable. End users temple visitors, event participants, needed clear navigation and accessible content. My mentor Jai supported me with guidance, and future developers would benefit from clean architecture and modular code.

Keeping everyone in mind helped me make decisions that were practical, user-focused, and sustainable. It reminded me that even a technical project is really about people.

Q7. Ethical Considerations

Restoring the site came with responsibilities. I took care to protect media integrity, validate file ownership, and avoid exposing any sensitive data. Where assets were missing, I used placeholders rather than misrepresenting anything.

Accessibility and fair access were priorities. I tested layouts, screen reader compatibility, keyboard navigation, and font/contrast choices to make sure the site is inclusive. I also optimized performance by removing unnecessary plugins and scoping CSS, which reduces load on servers and is a small way to consider environmental impact.

Navigation and interactions were designed to be transparent and no dark patterns, no misleading buttons. This ensures freedom of choice for visitors and builds trust.

The outcome is a site that's not only functional and visually polished, but also accessible, ethical, and responsible.

Q8. Stakeholder Collaboration Methods

Throughout the FGS legacy site restoration, collaboration wasn't just helpful it was essential. Even though I worked independently for most of the build, I actively engaged with stakeholders and mentors to shape the direction and ensure the final product met real needs.

One of the most effective practices was working together on shared documents. I used Notion to track weekly tasks, blockers, and QA issues. This board became a living document that I could share with my mentor and stakeholders it helped them understand progress, and helped me stay accountable. It also allowed for asynchronous feedback, which was crucial during busy weeks.

I also collaborated on the presentation itself. My mentor supported me in refining the structure and tone of my final slides, helping me balance technical depth with storytelling clarity. We discussed how to frame architectural wins, emotional growth, and deliverables in a way that felt authentic and confident.

There were moments where I reached out for help, especially during Week 7 when personal challenges made focus difficult. Communicating openly whether through messages or calls helped me stay grounded and reminded me that collaboration isn't just about code; it's about support and shared problem-solving.

Feedback loops were also key. I received input on layout decisions, shortcode logic, and accessibility fixes, and I folded that feedback into my Notion tracker and QA checklist. This

iterative process helped me refine the site in real time and ensured the final product was polished and inclusive.

In short, collaboration showed up in many forms shared documentation, presentation support, technical feedback, and emotional encouragement. It helped me respond to project requirements with clarity, adaptability, and a sense of unity.

Q9. Evaluating Communication & Collaboration Methods

The communication and collaboration methods I used during the FGS restoration were effective because they kept me grounded, accountable, and connected even while working independently.

Using Notion as a central hub was especially powerful. It allowed me to track tasks, blockers, and QA issues in real time, and share progress with my mentor and stakeholders. This asynchronous method worked well during busy or emotionally heavy weeks, especially when I was juggling personal challenges. It gave me space to reflect, document clearly, and stay aligned with project goals.

Collaborating on the presentation was another highlight. My mentor helped me shape the narrative, refine the tone, and balance technical depth with emotional honesty. That process helped me see the project not just as a technical build, but as a story worth telling.

Reaching out for help especially during Week 7 when things felt overwhelming was a turning point. Communicating openly helped me stay focused and reminded me that collaboration isn't just about code; it's about support, trust, and shared problem-solving.

What Could Be Improved

If I were to repeat this in a future project, I'd build in more structured feedback loops perhaps weekly check-ins or short calls to review blockers and wins. While asynchronous tools worked well, real-time conversations could have helped me catch issues earlier and feel more supported during high-pressure moments.

I'd also explore collaborative coding tools like GitHub Projects or live pair programming sessions for key components. That would allow for faster iteration and shared learning, especially when dealing with complex shortcode logic or database restoration.

Lastly, I'd make space for emotional check-ins even just a note in Notion to reflect on how I'm feeling. Because when the work gets heavy, having a space to acknowledge that makes the collaboration more human and sustainable.

Q10. Key Elements of My Contribution

My contribution to the FGS legacy site restoration was hands-on, architectural, and deeply personal. I didn't just patch up broken pieces I rebuilt the foundation with clarity, structure, and care.

One of the biggest elements was restoring the site from corrupted backups. I manually recovered media via FTP and Media Sync, validated gallery assets, and rebuilt key pages like Home, About, Contact, Temple Info, and Events using Gutenberg blocks. This required patience, precision, and a lot of troubleshooting.

I also designed and implemented modular CSS architecture, where each page or component had scoped styling. This made the frontend scalable and maintainable, and allowed future updates without breaking layout flow. I translated Figma designs into clean, responsive blocks, and created custom components like event cards and gallery carousels.

On the backend, I built a custom event system using CPT UI and ACF, with shortcode-based rendering and a custom archive template. This gave staff control over content without relying on bloated plugins, keeping the site fast and future-proof.

I created a Notion QA tracker to document device testing, layout issues, plugin behaviour, and accessibility fixes. This helped me stay organized and communicate progress clearly. It also became a key part of my handover documentation.

Beyond the technical work, I contributed resilience and reflection. I navigated personal challenges while staying focused, honest, and solution-oriented. I reached out when I needed support, communicated openly, and kept showing up even when things felt heavy.

In the end, I delivered a site that's functional, polished, and ready to grow. But more than that, I built trust in myself as a developer who can handle legacy systems, think architecturally, and lead with integrity.

Q11. Learnings and Future Employment Impact

This project taught me more than just technical skills it reshaped how I approach challenges, communicate under pressure, and think like a developer with long-term vision.

One major area of growth was my ability to troubleshoot legacy systems. I learned how to restore corrupted backups, recover media via FTP, and rebuild a WordPress site from the ground up using PHP, Gutenberg blocks, and custom post types. I now feel confident navigating broken environments and making sense of outdated architecture skills that are highly valuable in real-world dev roles.

Another key learning was architectural thinking. I didn't just style pages I built scoped, modular CSS that's maintainable and scalable. I translated Figma designs into reusable components,

created shortcode-based rendering logic, and documented everything clearly in Notion. This taught me how to build systems that others can understand, extend, and trust.

Just as important was learning how to self-manage and work completely solo. I planned, built, tested, and documented the entire restoration independently. I created my own QA tracker, structured my workflow in Notion, and made decisions without relying on constant oversight. That level of autonomy gave me confidence in my ability to lead projects, stay organized, and deliver results under pressure.

Beyond the technical, I learned how to communicate with clarity and integrity. Whether I was updating my mentor, logging QA issues, or presenting my work, I stayed honest about blockers and focused on solutions. That mindset owning the process, not just the outcome is something I'll carry into every future role.

These learnings directly enhance my employment prospects. I now have hands-on experience in WordPress restoration, scalable frontend workflows, and collaborative documentation. I've proven I can take a broken system and turn it into something clean, functional, and future-ready. And I've done it while managing real-life challenges, staying focused, and growing through every obstacle. I'm not just ready to contribute I'm ready to lead with clarity, resilience, and a genuine love for the craft.

Supporting Documentation

All supporting materials including:

10 Week Plan Updated,

10 Week Summary,

Blockers,

Events Setup,

Learning Log,

Notion QA Tracker

are included in my GitHub repository:
https://github.com/Cess-stack/fgs-nz-site-restoration.git

These files reflect my full restoration process from technical recovery and media restoration to architectural decisions and QA testing. They're organized in the `/docs` folder for easy access and future reference.