

GrewEstates - Custom WordPress Theme Development

by

Shaikh Radwan Ahmed Ratul (213014014)

*Internship report (CSE4099) submitted in partial fulfillment of the
requirements for the degree of*

Bachelor of Science in Computer Science and Engineering

Under the supervision of

Suravi Akhter

Lecturer, Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Abdullah Ziyad

Founder & CEO
GrewDev



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIVERSITY OF LIBERAL ARTS BANGLADESH

SPRING 2026

© *Shaikh Radwan Ahmed Ratul*
All rights reserved

DECLARATION

Internship Title	GrewEstates — Custom WordPress Theme Development
Author	<i>Shaikh Radwan Ahmed Ratul</i>
Student ID	213014014
Supervisor (Academic)	Suravi Akhter
Supervisor (Industry)	Abdullah Ziyad

I declare that this internship report entitled *Title of Internship* is the result of my own work except as cited in the references. The internship report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Shaikh Radwan Ahmed Ratul
213014014

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Date: April 21, 2026



Department of Computer Science and Engineering
University of Liberal Arts Bangladesh
Mohammadpur, Dhaka - 1207

CERTIFICATE

This is to certify that the internship report entitled **GrewEstates - Custom Word-Press Theme Development**, submitted by **Shaikh Radwan Ahmed Ratul** (Student ID: 213014014) is a undergraduate student of the **Department of Computer Science and Engineering** has been examined. Upon recommendation by the examination committee, we hereby accord our approval of it as the presented work and submitted report fulfill the requirements for its acceptance in partial fulfillment for the degree of *Bachelor of Science in Computer Science and Engineering*.

Suravi Akhter, Lecturer

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Abdullah Ziyad, Founder & CEO

GrewDev - Growth-Driven Web & Marketing Agency

Place: Dhaka

Date: April 21, 2026

ACKNOWLEDGEMENTS

I would like to express my deep and sincere gratitude to my industry supervisor, *Abdullah Ziyad*, Founder and CEO of GrewDev, for giving me the opportunity to conduct this internship and for providing invaluable guidance throughout the entire project. His clear technical direction, vision for quality, and willingness to mentor a junior developer made this experience genuinely transformative. Working under his leadership taught me not only how to write WordPress code but how to think like a professional developer accountable to real client expectations.

I am equally thankful to my academic supervisor, *Suravi Akhter*, Lecturer in the Department of Computer Science and Engineering at the University of Liberal Arts Bangladesh. Her consistent encouragement, timely feedback, and guidance on how to document and reflect on technical work have shaped both this report and my professional outlook.

I am greatly indebted to the honorable teachers of the Department of Computer Science and Engineering at ULAB who taught me during the course of my undergraduate study. Without their rigorous instruction in algorithms, software engineering, web technologies, and system design, I would not have had the conceptual foundation to absorb what this internship demanded of me.

I am extremely thankful to my parents for their unconditional love, their prayers, and their unwavering support through every semester of my degree. Their sacrifices made my education possible. I also thank my friends and classmates, whose companionship and collaborative spirit made the difficult parts of this journey bearable.

Finally, I would like to thank everyone at GrewDev who welcomed me into the team, answered my questions patiently, and gave me the space to learn by doing.

Shaikh Radwan Ahmed Ratul

University of Liberal Arts Bangladesh

Date: April 21, 2026

Dedicated to my family

– *Shaikh Radwan Ahmed Ratul*

EXECUTIVE SUMMARY

The following report is a documentation of an internship at GrewDev in Bangladesh under the tutelage of its Founder and CEO, Abdullah Ziyad. The internship was a period between February 10 and May 10, 2026 and the intern was assigned to the WordPress development team where he or she would be reporting to the Senior WordPress Developer.

The main reason behind this internship was to close the gap between theoretical understanding of PHP and nitty-gritty of real-world, production-scale web development. The goal was to work on a live client project - GrewEstates, a high-end custom WordPress theme that was developed into the real estate industry - and to have a hands-on experience with agency operations, standards of coding, and delivery of professional software.

The scope of work included the entire life cycle of the development of a seven-page WordPress theme, including Home, Properties, Property Detail, About Us, Contact Us, Find Agents, and a custom 404 page. The theme is designed to be generated without page builders, based on PHP 8.x, Bootstrap 5.3 and Vanilla JavaScript, to deliver excellent performance, search engine optimization and maintainable codebase.

The greatest success was the provision of a complete modular, production-ready theme that included five reusable PHP modules `property-card.php`, `agent-card.php`, `cta-banner.php`, `section-header.php`, and `breadcrumb.php` as well as conditional loading of assets that minimized redundant HTTP requests per page.

Such issues as learning the hook-based PHP paradigm of WordPress (which is not comparable to scholarly MVC frameworks), template hierarchy conflicts, a bug in the Customizer logo, and a responsive filter sidebar with a Vanilla JavaScript switch were also addressed. All challenges were addressed by reviewing documentation, debugging and mentoring by senior developers.

On the workplace competence, the internship nurtured skills in WordPress theme architecture, responsive UI design, escaping output to prevent XSS attacks,

and version control using Git. Personally, it provided a discipline of coding standards, DRY concepts, and intent-oriented documentation a robust base to work in scalable, full-stack web development.

Keywords: WordPress Theme Development, PHP, Bootstrap, Hierarchy

Contents

1	Introduction	1
1.1	Introduction	1
1.2	Objective of the internship report	2
2	Background	3
3	Scope of Work	5
4	Internship Activities	7
4.1	Introduction of the Organization/Institution	7
4.2	Work Accomplishment	8
4.3	Challenges Faced	19
5	Methodology	21
6	Analysis and Findings	23
6.1	Results and Findings	23
6.2	Learning and Development	24
6.3	Tools and Technology	25
6.4	User Guideline	26
6.5	Impact of the Internship	26
7	Conclusions and Recommendations	28
7.1	Conclusion	28
7.2	Limitations	29
7.3	Recommendations	29
8	Acknowledgments	31

Appendix	32
References	36

List of Figures

4.1	phpMyAdmin showing grewestates _{db}	9
4.2	WP Admin Themes page showing GrewEstates active	10
4.3	style.css theme header in VS Code	11
4.4	functions.php requiring all inc/ files	12
4.5	Properties page with filter sidebar visible	13
4.6	Property Detail page	13
4.7	WP Customizer panel open with GrewEstates settings	14
4.8	About Us page	14
4.9	Find Agents page	15
4.10	Contact Us page	15
4.11	404 page	18
5.1	inc/enqueue.php showing Bootstrap 5.3 conditional enqueue	22
6.1	Home page rendered in browser	23
6.2	Bootstrap 5.3 in browser Network tab	25
6.3	WP Admin Pages list with all 7 pages and templates assigned	26

List of Tables

Chapter 1

Introduction

1.1 Introduction

The web development sector in Bangladesh is now moving away to generic templates to high-performance, custom-built systems which provide more control over brand identity. Within such a competitive environment, WordPress is still a powerhouse, driving a large percentage of the digital economy. As a growth-driven agency, GrewDev prioritizes custom WordPress theme development to eliminate technical “bloat” while ensuring superior SEO, security, and performance. Themes developed by ground-up engineering allow developers to apply particular design tokens, like the luxury gold-and-dark palette, without losing the ownership of the PHP architecture and component-based structure.

During this internship, I developed GrewEstates, a high-quality real estate WordPress theme that was manually built. Its front-end was a full-fledged project, which included creating a seven-page architecture. This architecture included dynamic home and property listings, detail pages for each property, About Us, Contact Us, Find Agents, and a custom-styled 404 page. The project was built on a technical stack of PHP 8.x, Bootstrap 5.3 [Bootstrap \(2024\)](#), and Vanilla JavaScript, and aimed to provide a functional tool for real estate professionals, with improved property cards, pages for expert agents, and a modern layout that would be easy to use in modern browsers.

The internship report covers a 10-week intensive development phase from February 10, 2026 to April 15, 2026. During this period, the report covers the entire lifecycle of the GrewEstates project, including the initial setup of the WampServer [WampServer \(2024\)](#) and Windows 11 environment and finally the addition of

multi-column menus and template page layouts. The following sections provide a technical description of the methods used to bring the site to professional agency standards.

1.2 Objective of the internship report

The main aim of the internship report is to give a detailed end-to-end documentation of the steps followed in creating a production-grade, custom WordPress theme [WordPress.org \(2024c\)](#). This involves explaining how the stationary HTML/CSS elements will be converted into dynamic PHP template files, such that all features, including the navbar scroll state being controlled by the `requestAnimationFrame`, and the price slider labels being interactively responsive, will work correctly in the WordPress core ecosystem. The report will be considered as a written document of how I was able to apply agency level design standards yet met the technical limitations of a high performance web environment.

Additionally, the report also seeks to show technical expertise in integrating Bootstrap 5.3 [Bootstrap \(2024\)](#) using standardized enqueueing, manipulating the WordPress loop, and utilizing principles of responsive design using multiple device breakpoints. The report fills the gap between theoretical understanding of PHP and the real-world development processes that necessitate development driven by growth that is needed in a professional digital agency. Finally, this document points out the personal and professional development that has occurred throughout the time I worked at GrewDev, which will enable me to pursue a career in creating scalable, innovative software solutions.

Chapter 2

Background

GrewDev is a specialized and growth-driven web and marketing agency. The company is concerned with providing high performance digital solutions by combining high web development and strategic marketing objectives. GrewDev works with a wide range of local and global customers. The company has a team structure where interns collaborate with the experienced designers, and senior WordPress developers. This is an environment that will help promote professional growth by involving itself in the current projects of the agency, to ensure that all deliverables are of the best industry standards in terms of quality and performance, under the leadership of Founder and CEO Abdullah Ziyad.

The online needs of the real estate business in Bangladesh and all over the world have developed into a multifaceted ecosystem that needs property listing websites, agent directories, and advanced lead generating software. Modern real estate companies use one of the most trusted web sites to close the communication gap between real estate brokers and potential buyers of the home. GrewEstates was opened as a client project to meet those targeted market needs by offering a digital marketplace that is centralized and high-end. The project serves as an effective tool for real estate professionals, featuring dynamic property search, filterable listings, and detailed profiles of each property. By automating the property search process, the platform aims to reduce the complexities inherent in traditional real estate transactions and provide a streamlined experience for both administrators and end-users.

For a high-profile project like GrewEstates, it was preferable to build a custom WordPress theme WordPress.org (2024c) instead of using a generic page builder like Elementor. Page builders can be used to quickly prototype, but can be prone

to technical bloat which slows down the speed of the website and search engine optimization. Creating the theme in a component-based PHP structure and starting with plain code, GrewDev retains all rights over the code and has a better control over the logic behind it. This practice assists in applying particular design tokens, e.g. the premium gold-and-dark root palette as laid out in the core CSS of the theme. Also, a custom built theme will be more maintainable and scalable to subsequent stages of the project and will enable complex features to be added without the constraints or security concerns that may come with third-party visual editors.

During this ten-week period, I worked as a Junior WordPress Developer Intern and reported directly to the Senior WordPress Developer. My tasks were diverse in technical tasks typical of the workflow of a professional agency. According to my internship contract, I was to apply best practices in theming customization, integrating a few plugins, and security of websites [WordPress.org \(2024b\)](#). I was engaged in tasks such as optimizing websites and quality assurance testing every day. I also had to do frequent bug fixing so that the site could be functional and user-friendly. These roles allowed me to get practical experience in all aspects of the development lifecycle, including setting up an environment and deploying final content.

GrewEstates has been built upon some fundamental engineering concepts and theories that underpin its technical development. The structure of the project is based on the WordPress template hierarchy [WordPress.org \(2024c\)](#) and has dedicated PHP files of the home, property listing, and property detail pages. PHP hooks and a filter system are used to maintain modularity in the backend logic and prevent changes to the core files [WordPress.org \(2024a\)](#). The frontend is based on the grid system of Bootstrap 5.3 [Bootstrap \(2024\)](#) to achieve a fully responsive design and the CSS structure is based on the Block Element Modifier (BEM) methodology to provide consistency and readability of the class names. The interactive features are developed with vanilla JavaScript to have a light footprint. All development was done in a WampServer [WampServer \(2024\)](#) local environment on Windows 11, which facilitated a professional iterative cycle of coding, testing, and debugging.

Chapter 3

Scope of Work

The scope of work for the GrewEstates project included the complete design and engineering of a premium, fully custom WordPress theme [WordPress.org \(2024c\)](#) for a seven-page real estate platform. The ten-week experiment involved converting high-quality, Figma-quality design into a usable and responsive digital marketplace. This development cycle focused on agency-quality execution and structural integrity and made more than mere customization of templates to create a custom codebase directly made to develop growth-driven marketing. The primary objective of this scope of work was to offer a high-performance user interface with the backend being modular to be extended to administration in future.

The theme has an architectural design of a modular PHP design that is structured to have maximum maintainability and clarity. The `functions.php` file is at the heart of the system and it serves as the central engine as it needs specialized logic files found in the `inc/` directory. This is a collection of `inc/setup.php`, `inc/enqueue.php`, `inc/menus.php`, `inc/widgets.php`, and `inc/customizer.php`. Moreover, the project implements the `template-parts/components/` directory to store reusable UI fragments and `page-templates/` directory to contain the logic specific to the seven scoped pages [WordPress.org \(2024c\)](#).

GrewEstates asset management is predetermined by a highly hierarchical structure that appreciates optimization based on a conditional enqueueing approach [WordPress.org \(2024a\)](#). Defined in the `assets/css/` directory, the `main.css` file handles global root variables and typography, while page-specific styles such as `properties.css` or `home.css` and page-specific scripts such as `main.js` are managed through `inc/enqueue.php`. The theme will only include those CSS and JavaScript files that it actually requires at the given time to support the template it

is using, which will radically reduce unnecessary HTTP requests and improve page load times.

Much of my development effort was on the creation of reusable PHP pieces in the `template-parts/components/` directory. I have made five files to this end: `property-card.php` is used to handle the regular property listing, `agent-card.php` is used to handle the agent directory, and `breadcrumb.php` is used to handle the interior page navigation. I also created the call-to-action sections in marketing via `cta-banner.php` and common page titles via `section-header.php`. I do not have to repeat the same HTML structure five times in various templates, but only once and refer to it in any place as necessary. When something is changed, I update just one file and it is reflected in all places. Prior to this internship, my understanding of the DRY principle had been limited to university-level study but I had never put the concept into practice on this level in a project. This also facilitated the maintenance of the project. At the very beginning, I had to perform a small styling modification in the property card layout and manually update 3 template files. Once I had everything refactored into the component system, the same type of change only took around two minutes. In retrospect, I ought to have established this structure earlier on, as it proved to be a valuable learning experience of how a real codebase ought to be structured.

Chapter 4

Internship Activities

4.1 Introduction of the Organization/Institution

GrewDev is a growth-oriented and professional web and marketing agency. The company is about delivering high performance digital solutions through integrating high web development and strategic marketing goals. GrewDev has a broad palette of customers locally and internationally. The company has a team structure in which interns work with the senior designers, and the more experienced WordPress developers. This is a workplace that will contribute towards the growth of the professionals by engaging itself in the ongoing projects of the agency, to make sure that all the deliverables are of the best standards in the industry in terms of quality and performance under the leadership of Founder and CEO Abdullah Ziyad.

GrewDev has a large portfolio of clients and the primary ones include, but are not limited to, developing WordPress themes WordPress.org (2024c), WooCommerce integration, and performance-based marketing platforms. The interesting aspect of the internship format to me was that the interns are not allocated practice tasks in isolation. I was tasked with assisting senior developers in frontend and backend development, conducting testing of the websites, and observing the internal security practices that the team had observed. This was not as I had anticipated. I assumed that during the initial weeks of the internship I would be primarily an observer, but the GrewDev staff anticipated interns to be productive at the outset. Due to this arrangement, I was able to experience the entire project lifecycle. I participated in the early environment configuration to the last deployment process. It is difficult to acquire that type of exposure in a university environment. In retrospect, it was

difficult to be exposed to the actual client work at such a rapid rate, but it is also what gave me the knowledge I acquired at this time.

4.2 Work Accomplishment

Week 1

Week 1 was primarily preparation. The primary activity I did was to prepare the local development environment and start WordPress preparation in the GrewEstates project.

- WampServer [WampServer \(2024\)](#) is used by the agency in Windows 11 to develop locally. The concept is to have all things in isolation prior to anything approaching a live server.
- Onboarding helped to understand that there are common practices when it comes to naming of databases and even the initiation of themes. I was not anticipating such organisation during the first day, but it was logical when I found out the way the team operated.
- I installed WampServer and WordPress, created the database `grewestates_db` using phpMyAdmin and configured `wp-config.php` with appropriate database credentials. Then I created the theme folder and added the `style.css` header with Theme Name: GrewEstates and Author: GrewDev, which is what WordPress uses to identify a theme [WordPress.org \(2024c\)](#).
- I also learned how to set the permalinks to Post Name in the dashboard. This provides clean, SEO-friendly URLs as opposed to the default `?p=123` format. I then activated the custom theme, which at this stage was a fairly blank folder.
- The most significant conceptual change was the realisation that WordPress handles PHP differently from what I had been taught in the classroom. At university we worked with frameworks such as Laravel in which Model, View and Controller are kept separate. WordPress does not enforce that. It relies on a file-based template hierarchy [WordPress.org \(2024c\)](#) with logic and presentation frequently co-located in a single PHP file. That was not easy to get used to.

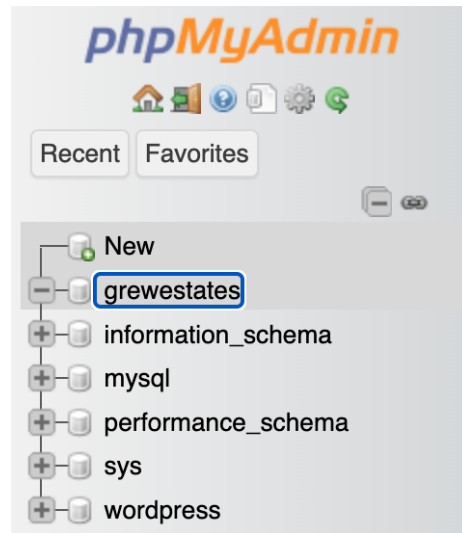


Figure 4.1: phpMyAdmin showing $grewestates_{db}$

Week 2

This week was dedicated to the construction of the theme core architecture and the establishment of an appropriate asset management system.

- Professional themes do not put everything in `functions.php`. Rather, they adopt a modular include system to keep things clean and easy to maintain.
- Among the initial steps, I created design tokens for the brand — `ge-gold` (`#C9A84C`) and `ge-dark` (`#1A1A1A`) — in the `:root` of the CSS. In this manner the colours are managed at a single location.
- I established the `inc/` directory and created `setup.php` for `add_theme_support()` calls and `enqueue.php` to manage assets. I used `wp_enqueue_style()` and `wp_enqueue_script()` [WordPress.org \(2024a\)](#) to load Bootstrap 5.3 [Bootstrap \(2024\)](#), Bootstrap Icons, and Google Fonts Inter [Google Fonts \(2024\)](#). I also used `register_nav_menus()` to register the primary navigation menus in `inc/menus.php`.
- Prior to this, I had simply hardcoded `<link>` tags into HTML. I learned that it is preferable to enqueue assets through WordPress [WordPress.org \(2024a\)](#) since dependencies and versioning are automatically managed, and it prevents the same library from being loaded twice.

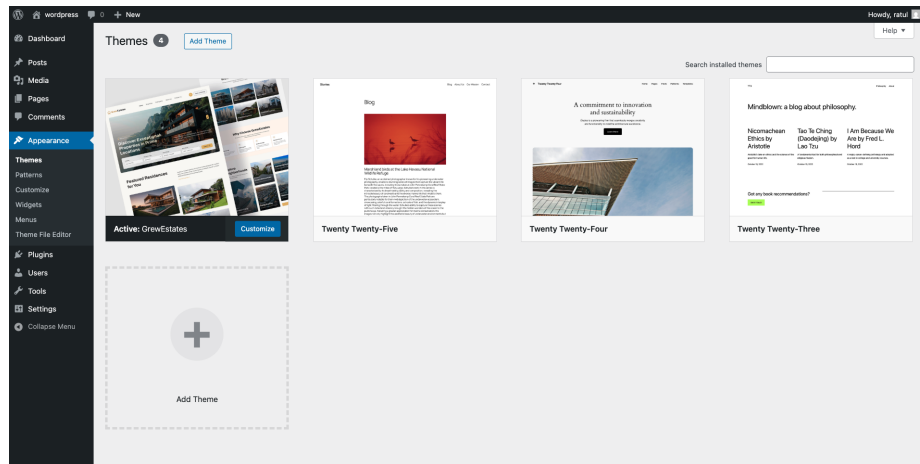


Figure 4.2: WP Admin Themes page showing GrewEstates active

- I also came to know that `header.php` and `footer.php` must contain `wp_head()` and `wp_footer()` respectively [WordPress.org \(2024a\)](#). Those hooks are needed for plugins to inject their scripts and styles into the page. I did not notice this initially and discovered it when one of the plugins was not working.

Week 3

This week I moved into developing the core template files that comprise the WordPress template hierarchy [WordPress.org \(2024c\)](#).

- I learned how WordPress determines the PHP file that it loads. As an example, it will select `single.php` for a single post and `archive.php` for a listing page, both depending on the URL requested.
- It was more difficult than I thought to convert static HTML into dynamic PHP templates. The length of content varies and that must be considered in the layout.
- I created the dominant hierarchy files: `index.php`, `page.php`, `single.php`, `archive.php`, `404.php`, and `search.php`. I applied `get_header()`, `get_footer()`, and `get_template_part()` to prevent repetition of code (DRY principle). I also developed the `global navbar.php` and placed it in `template-parts/global/`.

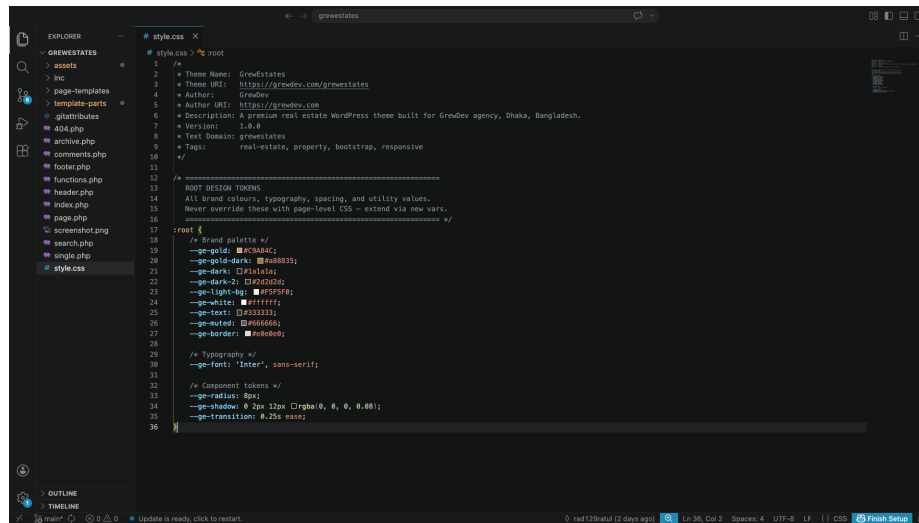


Figure 4.3: style.css theme header in VS Code

- I had to get my head around the Loop. By combining `have_posts()` and the `_post()`, WordPress iterates over the database query and displays each post [WordPress.org \(2024a\)](#). Once I understood it, most other things fell into place.
- I also discovered the `page-templates/` folder [WordPress.org \(2024c\)](#). Placing a comment at the top of a PHP file with a template name causes WordPress to display it as an option in the editor, allowing admins to give individual pages their own layout without any code handling.

Week 4

The primary task of this week was to develop the Home page — transforming the design into a functional WordPress template.

- There was a lot going on the Home page. It required a Hero section, Featured Residences, a Neighbourhoods section and so on. Planning all of that within a single template required careful thought.
- I used Vanilla JavaScript to add certain UI behaviours, namely the search bar interaction and the scroll effect on the hero section. This was new to me in the context of WordPress.

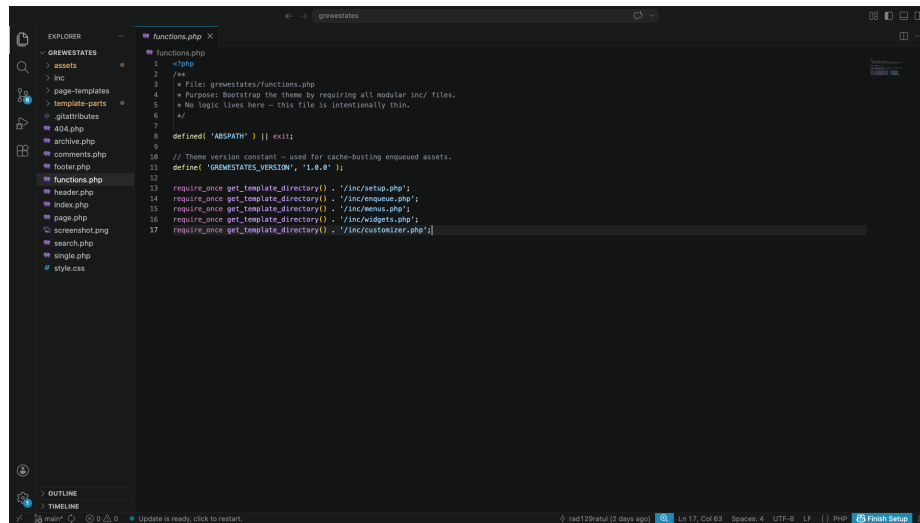


Figure 4.4: functions.php requiring all inc/ files

- I created `template-home.php` with sections for Browse by Property Type, Why Choose GrewEstates, and Expert Agents. I created `home.css` for page-specific styles and built `property-card.php` and `agent-card.php` as reusable elements within `template-parts/components/`.
- I learned how to use `get_query_var()` to pass data to those components [WordPress.org \(2024a\)](#), for example passing the property card variant for residence grids.
- The key lesson was why componentisation is important. I only had to write `agent-card.php` once and reuse it on both the Home page and the Find Agents page. This proved to be a significant time-saving measure.

Week 5

This week I worked on the Properties listing page. It required a precise Bootstrap 5.3 layout and a functioning filter mechanism, making it one of the more complex pages to date.

- Listing pages must load data fast and also provide the user with enough filters to find what they want. Getting that balance right required careful thought.

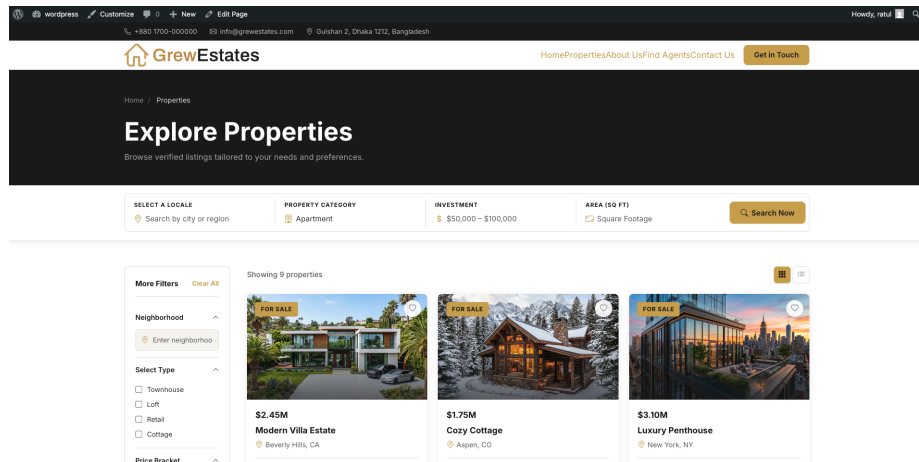


Figure 4.5: Properties page with filter sidebar visible

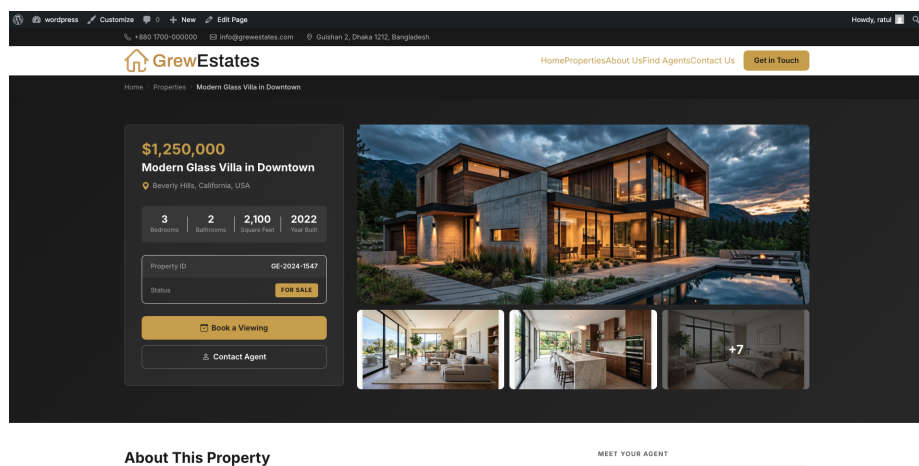


Figure 4.6: Property Detail page

- The sticky sidebar was more difficult than anticipated. Ensuring it remained functional across screen sizes without overlaying the main content area required several iterations.
- I created `template-properties.php` based on a Bootstrap grid [Bootstrap \(2024\)](#) comprising a `col-lg-3` sidebar and a `col-lg-9` results container. I implemented `breadcrumb.php` and `properties.js` to handle the sidebar toggle and switching between grid and list views. I also updated the conditional logic in `inc/enqueue.php` to ensure that `properties.css` and its script are only loaded on this particular template.

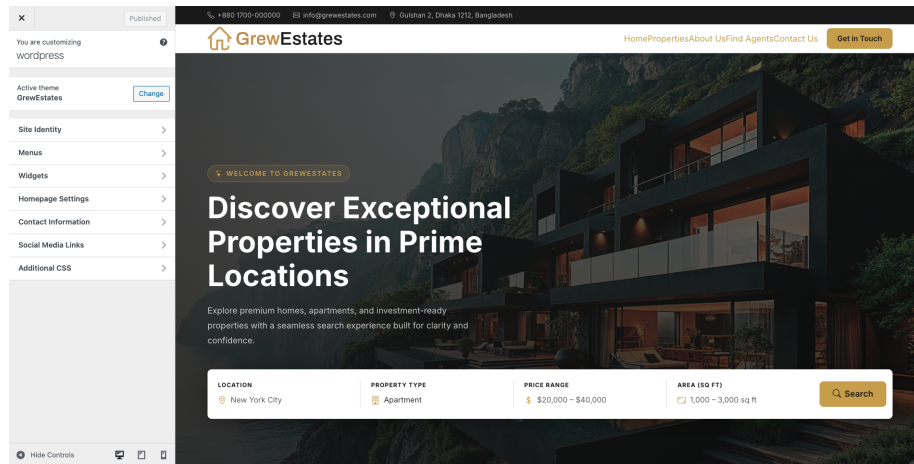


Figure 4.7: WP Customizer panel open with GrewEstates settings

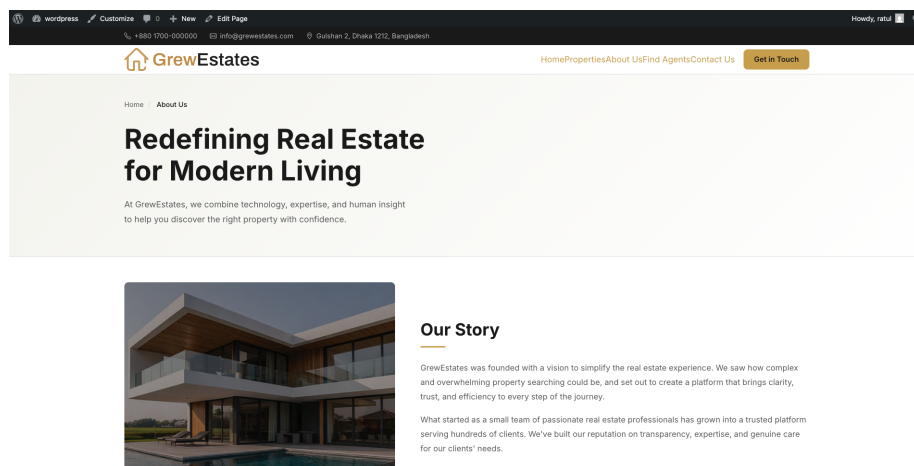


Figure 4.8: About Us page

- I learned how to implement client-side pagination with Bootstrap classes **Bootstrap (2024)**. This avoids cluttering the UI with a large number of properties and avoids a server round-trip on each page change.
- The broader lesson was asset scoping. Separating page-specific styles keeps the global theme lightweight, as each page loads only what it requires.

Week 6

This week covered two aspects: implementing the WordPress Customizer API and developing the single property view.

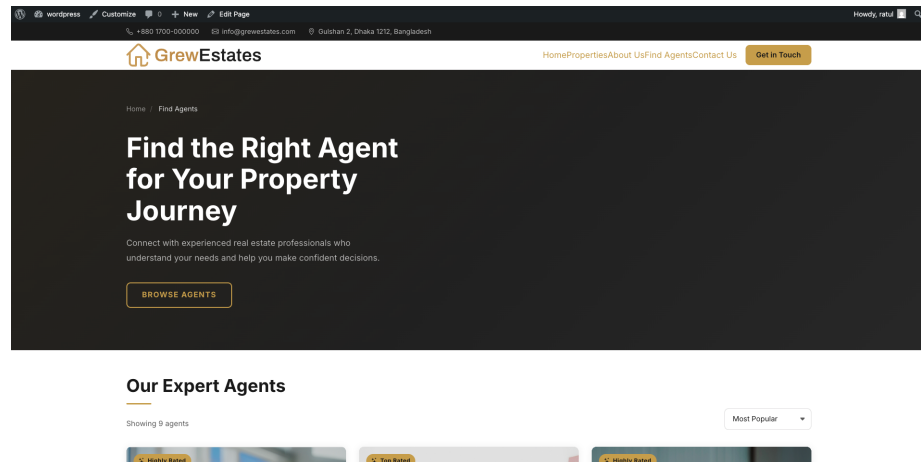


Figure 4.9: Find Agents page

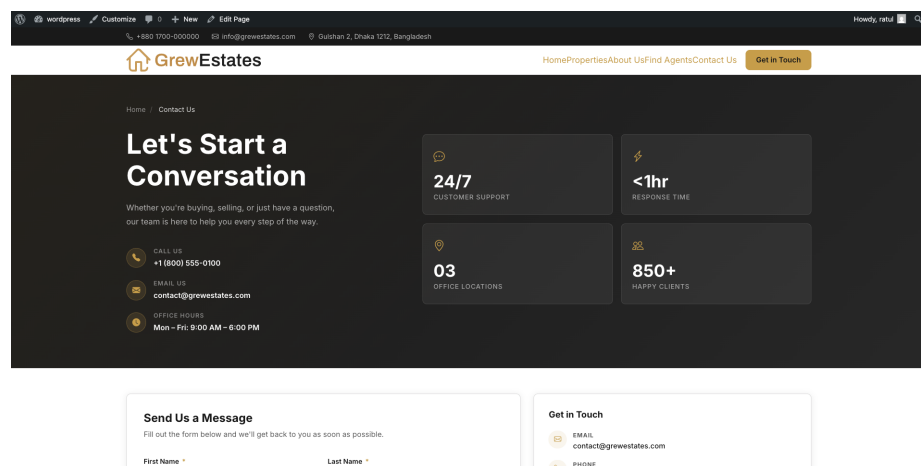


Figure 4.10: Contact Us page

- I had not used the Customizer before. It allows admins to modify theme settings and view the outcome live, which is convenient for non-technical users.
- The postMessage transport for live previews was new to me. It does not reload the entire page to update the preview, which provides a much smoother experience for the administrator.
- I used `add_panel()`, `add_section()`, `add_setting()`, and `add_control()` to configure `inc/customizer.php` [WordPress.org](https://wordpress.org) (2024a). This enabled uploading logos, updating contact information, and managing social links through

the dashboard. I also created `template-property-single.php`, displaying dynamic pricing, property details, and a dedicated agent sidebar.

- I learned to distinguish between `get_theme_mod()` and `get_option()` [WordPress.org \(2024a\)](#). `get_theme_mod()` is used for theme appearance options while `get_option()` is used for global site data. I had previously used them interchangeably, which was incorrect.
- The single property page is the most critical page of a real estate site. The design must lead with quality images and expose the required call to action clearly. I kept that in mind throughout its construction.

Week 7

This week focused on the About Us and Find Agents pages, and marked a turning point in my understanding of WordPress PHP.

- The About page contained narrative elements such as Our Mission and the Leadership Team. These required particular UI patterns — floating badges, card grids, etc. — to look polished rather than resembling a wall of text.
- Building the Find Agents directory with nine responsive agent cards filtered using CSS tabs was more difficult than anticipated. Getting the tab logic to work without JavaScript was a good challenge.
- I prepared `template-agents.php` and `template-about.php`, along with `cta-banner.php` and `section-header.php` to keep sections visually consistent. I also included a CSS counter animation in the stats section to add life to the page without the overhead of JavaScript.
- This was the week I properly understood `add_action()` and `add_filter()` [WordPress.org \(2024a\)](#). These hooks allow logic to be injected into WordPress without altering core files, which is quite unlike the procedural PHP I had written at university.
- The paradigm shift here was realising that WordPress PHP is hook-based, not procedural. Code is not run sequentially — WordPress executes it at the appropriate time. This was not something academic PHP had prepared me for at all.

Week 8

Week 8 focused on the Contact page and the site-wide footer.

- The footer is more significant than it appears. It carries secondary navigation and brand information and must be consistently structured across the entire site.
- The bulk of my time was spent writing form validation in Vanilla JavaScript. Handling various input types such as phone numbers and email formats without pulling in a library required careful logic.
- I created `template-contact.php` including a stats bar, office locations, and an FAQ section. I developed `contact.js` for client-side validation and configured a 4-column footer in `footer.php` and `footer-content.php` with a brand section, Quick Links, Property Types, and Contact Info.
- I learned how to use `wp_nav_menu()` [WordPress.org](https://wordpress.org) (2024a) to pull footer Quick Links from menus managed in the WordPress dashboard, meaning the client can update those links without touching any code.
- Contact forms are not only a UI element. They serve as the primary lead-generation mechanism on a site of this type and must provide clear visual feedback upon submission.

Week 9

Week 9 was dedicated to QA — ensuring the theme was secure, fast, and free of apparent bugs.

- A good 404 page is more than an error message. It can redirect users to useful sections of the site and salvage a visit that would otherwise be wasted.
- Cross-browser testing was more tedious than anticipated. There were minor layout variations between Chrome and Edge that each needed to be found and fixed individually.

- I created 404.php with a popular pages grid and 404.css to style it. I also audited the codebase to ensure all dynamic outputs were escaped using `esc_html()`, `esc_url()`, and `esc_attr()` [WordPress.org \(2024b\)](#). Finally I produced the `screenshot.png` theme thumbnail (880×660px) for the WordPress Admin dashboard.
- I came to understand that output escaping is not optional at the agency level [WordPress.org \(2024b\)](#). WordPress provides specific functions for this purpose and their consistent application is what separates a secure theme from a vulnerable one.
- Performance is not only about raw speed but also perceived speed. Conditional asset loading means a user browsing the agent directory does not need to download the CSS for the contact form.

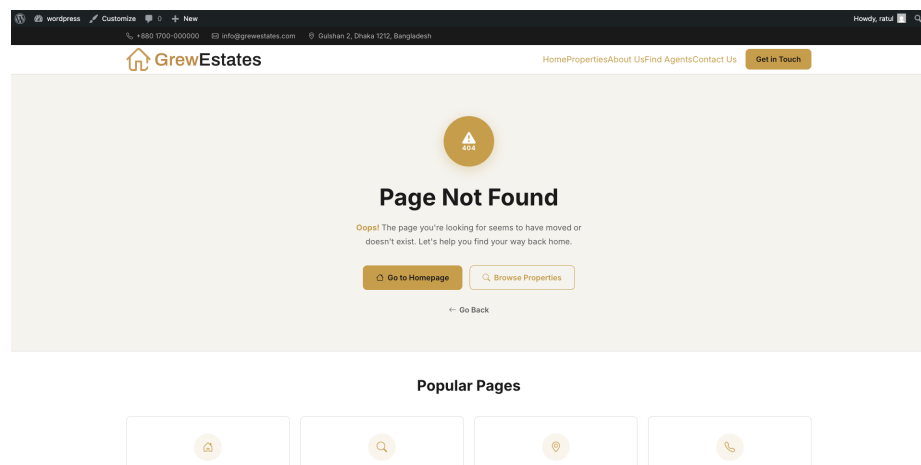


Figure 4.11: 404 page

Week 10

Week 10 was dedicated to documentation, cleanup, and formal handover to the GrewDev senior development team.

- Clear, intent-based comments in `functions.php` are more valuable than I had previously appreciated. They enable another developer to resume work on the project without needing to ask questions.

- The formal code review was a valuable experience. My code was compared against the WordPress Coding Standards [WordPress.org \(2024b\)](#) and I received direct feedback on what could be improved.
- I cleaned up the theme by removing unnecessary code and updated `directory-map.md` to reflect the final file structure. I completed the handoff with the Senior WordPress Developer, ensuring all dependencies were enqueued and documented.
- Documentation and standards compliance represent the final ten percent of a project. They usually determine its long-term maintainability.
- In retrospect, there is a clear gap between academia and professional practice. University provided me with the foundations of logic and problem solving. GrewDev provided me with the security mindset, coding standards, and the discipline that teamwork demands in real software engineering.

4.3 Challenges Faced

Challenge 1 — WordPress Enqueue System I thought the first step was to code the Bootstrap `<link>` tag into `header.php`. It was functional locally, but during one of the code reviews, the senior developer noted that, in this way, it had bypassed the asset management of WordPress completely. The appropriate hook is `wp_enqueue_scripts` [WordPress.org \(2024a\)](#). I transferred it all to `inc/enqueue.php`, which correctly handles dependencies and avoids conflicts when multiple plugins attempt to load the same library.

Challenge 2 — Template Hierarchy Confusion I did not completely comprehend how WordPress determines the template file to be loaded [WordPress.org \(2024c\)](#). At some stage a custom layout was not being displayed at all when I was developing locally and I could not understand the reason. Reading the official hierarchy documentation and experimenting with various file combinations helped me understand that `page.php`, custom templates, and `index.php` are in order of precedence over each other. Once I began to use the `page-templates/` folder properly, controlling special page layouts became straightforward.

Challenge 3 — Customizer Bug The logo appeared perfectly in the Customizer preview but not on the frontend. That was confusing. The debugging process helped to identify two issues: a syntax error in `inc/customizer.php` and the use of the deprecated `get_bloginfo('template_directory')`. I updated the settings parameter in `add_control()` and changed to `get_template_directory_uri()` [WordPress.org \(2024a\)](#), which resolved the image path.

Challenge 4 — Responsive Sidebar on Properties Page The filter sidebar on the Properties page would not collapse on mobile. It was violating the Bootstrap grid [Bootstrap \(2024\)](#) — `col-lg-3` and `col-lg-9` were crossing each other rather than stacking. I resolved the issue by correcting the grid classes and creating a Vanilla JavaScript toggle in `properties.js`, which makes the sidebar full-width at smaller screens.

Challenge 5 — Academic PHP vs. WordPress PHP This was the most difficult adjustment. PHP as taught at university is a top-to-bottom language, with logic written and executed in order. WordPress does not operate that way. All logic passes through `add_action()` and `add_filter()` [WordPress.org \(2024a\)](#), and code is only run when it is invoked at the appropriate stage of the lifecycle by WordPress. I studied hook priorities and execution order until I was able to inject logic into the theme without touching core files.

Chapter 5

Methodology

The theme development of GrewEstates began with the analysis of design and was written after. The agency also supplied the files of the design in the PDF and SVG formats, and I browsed each of the files in separate sections to define common patterns and layout demands. This assisted me in designing early the template-parts and global structures. This initial investment saved a lot of work in the future. It is what the GrewDev design team had defined in those files that formed the HTML structure that I later constructed.

Avoiding repeated code was one thing that I paid attention to initially. In the design audit, I identified some of the UI blocks that were repeated on several pages. The `property-card.php` module, such as, was required on the Home page and Properties page. The `agent-card.php` showed up in the Expert Agents section and the Find Agents directory. The `cta-banner.php` was implemented throughout the pages of About, Agents, and Contact. I have developed all these as individual files within `template-parts/components/` and have not touched the page templates. With `get_query_var` being used to send dynamic data such as price badges or agent ratings to these components, a single CSS change in `property-card.css` would apply to all the instances.

To perform, I applied conditional loading of assets in `inc/enqueue.php` rather than loading all the assets on each page. This was made possible by the use of the `is_page_template()` logic and other WordPress conditional tags. Global styles such as `--ge-gold: #C9A84C` and the Inter font-family were loaded on all pages, whereas things such as `assets/css/pages/properties.css` and `properties.js` were only loaded when their templates were used. This made the theme light and enhanced Time-to-Interactive to the users. Such features as the price slider or the

navbar scroll state that is controlled by `requestAnimationFrame` continued to be functional, as their scripts were loaded precisely when they were required.

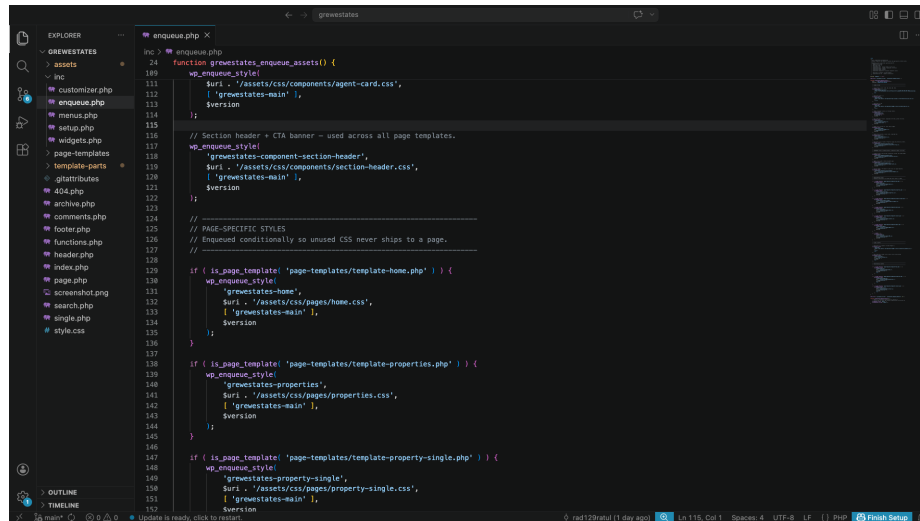


Figure 5.1: `inc/enqueue.php` showing Bootstrap 5.3 conditional enqueue

The construction was in a logical progression. Weeks one and two discussed the world shell — WampServer setup, basic theme support `inc/setup.php` and making `header.php` and `footer.php` play with hooks. The third week was on achieving the hierarchy of the WordPress template whereby, `index.php`, `page.php` and `single.php` were all identified accordingly. The seven major pages (Home and Properties) were discussed in weeks four to ten. The sequential order of the building allowed common elements such as `breadcrumb.php` and `section-header.php` to be completed before the interior pages required it.

I stuck to WordPress Coding Standards to ensure that the theme is safe and interoperable. All dynamic output was sanitized — `esc_html()` on text, `esc_url()` on links, `esc_attr()` on HTML attributes. This is resistant to XSS vulnerabilities. Any custom functions and variables are prefixed with `grewestates_` to ensure that they do not conflict with any plugins. I did not use inline style, hard-coded URLs and commented-out code either. A future developer working in an agency must be capable of opening the codebase and reading it without making educated guesses. That is what I attempted to keep myself to.

Chapter 6

Analysis and Findings

6.1 Results and Findings

The key deliverable of this internship was that seven fully functional WordPress page templates have been developed during the 10 weeks of the GrewEstates project. These include the Home, Properties, Property Detail, About Us, Contact Us, Find Agents and even the custom 404 page — all designed to the standards of the design of the agency. This was facilitated by developing five reusable components at the start: `property-card.php`, `agent-card.php`, `cta-banner.php`, `section-header.php`, and `breadcrumb.php`. By storing them in `template-parts/components/` they meant I did not need to recreate the same HTML in several templates, and visual consistency remained the same throughout.

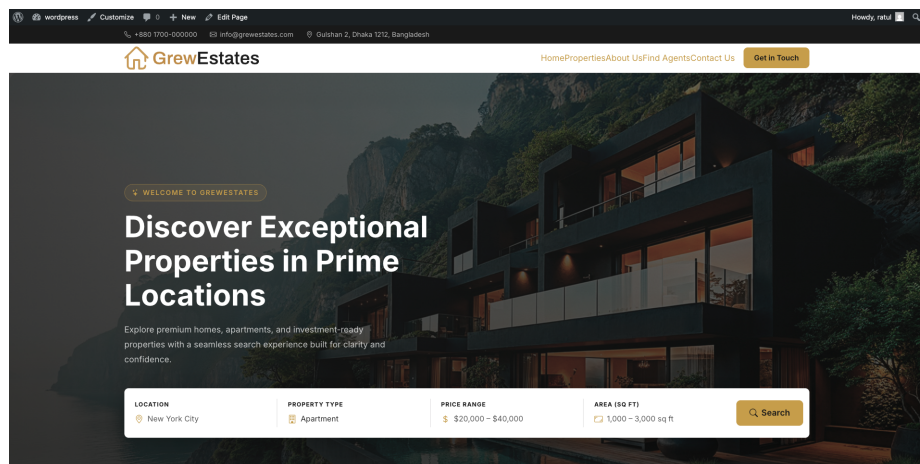


Figure 6.1: Home page rendered in browser

The theme got so far that all that is needed in deploying it is to install it and to activate the `grewstates` folder. The file structure is well-organized: `main.css` contains global design tokens and Bootstrap overrides, and seven per-page CSS files are located in `assets/css/pages/`. Three page-specialised scripts — Home for hero, Properties for filter toggling, and Contact for form validation — were required to create the pages on the JavaScript side. Viewing the end product, I believe the theme is not only a site, but a clean and lightweight software asset that can be easily handed over or expanded by GrewDev.

6.2 Learning and Development

This internship provided me with a practical opportunity to understand how professional development of WordPress differs from what we do at university. The greatest change was getting familiar with the hook system. When working with MVC frameworks in academic PHP, we write direct-call-function code or manipulate MVC frameworks. WordPress does not do so. You enrol your logic with `add_action` and allow the system to invoke it at the appropriate moment — such as the enqueueing of assets by `wp_enqueue_scripts` takes care of all asset enrolment. As soon as I realised this, a lot of things fell into place. Prior to that, I was merely hardcoding things and wondering why best practices said not to.

I became a lot more familiar with how to incorporate Bootstrap 5.3 into a PHP environment properly as well. Rather than add a `<link>` tag to the header, I passed the dependency array with `wp_enqueue_style()` so Bootstrap is loaded initially, and `main.css` loaded afterwards — the way CSS cascading is meant to operate. I applied BEM naming and utility classes of Bootstrap on the front-end side. A component may then include a class such as `.property-card__price` to be styled in a custom way and `card-body` to be formatted by Bootstrap. It required some practice to keep these two systems apart, but towards the end I was able to do it without much thought.

The final lesson I learned was a due respect to coding standards within a team. I was taught to delete the so-called zombie code — the commented-out blocks that accumulate and clog up any file read later on. I began to write intent-based comments in `functions.php` (explaining why something is done, rather than what it does). I also became strict on output escaping, always using `esc_html()` and

`esc_url()` to prevent XSS. These are small habits but in retrospect they are what makes code that works different from code that can be maintained.

6.3 Tools and Technology

The GrewEstates theme was developed using WordPress 6.x as the base CMS, with PHP 8.x handling the template hierarchy and database communications. To develop it locally, I worked with WampServer 3.x on Windows 11. phpMyAdmin 5.x was used to manage the database, with me creating and configuring the `grewestates_db` and SQL tables in WordPress. My IDE was Visual Studio Code, where I edited the PHP files in the `inc/` directory and managed the CSS structure.

I applied Bootstrap 5.3 for the responsive grid on the front end and Bootstrap Icons 1.11 for UI elements. Every interactive component was coded in Vanilla JavaScript (ES6+) — such as the navbar scroll state using `requestAnimationFrame` in `main.js` and the price slider logic in `properties.js`. I consciously avoided jQuery to keep the theme lightweight. Google Fonts Inter was used for typography to maintain a professional appearance. Git and GitHub served as version control, and Chrome DevTools were open nearly all the time for debugging CSS and checking JavaScript event listeners in real time.

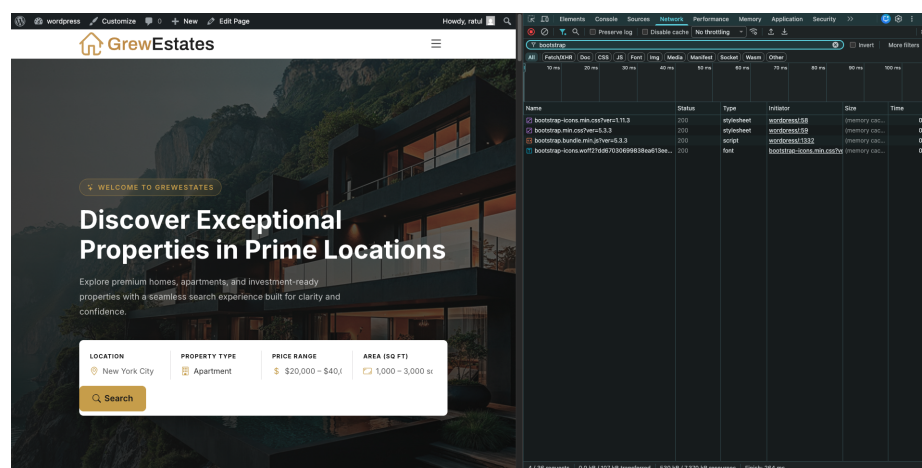


Figure 6.2: Bootstrap 5.3 in browser Network tab

6.4 User Guideline

The GrewEstates theme requires a fresh installation of WordPress 6.x on a compatible server. Then, drag and drop the `grewestates/` folder into `wp-content/themes/`. Once it is there, activate it in the WordPress administration panel under **Appearance > Themes**. The following step is critical — under **Settings > Permalinks**, select **Post name** in order to have the custom template routing running properly.

Next, create five pages in the Pages section: Home, Properties, About, Contact and Find Agents. For the Home, Properties, and Agents pages, select the appropriate template in the **Template** dropdown of the right sidebar. Then visit **Settings > Reading** and set the homepage to a static page, selecting the Home page you created. Finally, create the Primary Menu under **Appearance > Menus** and assign it to the Primary Navigation location. The WordPress Customizer also allows you to upload a logo and update the agency contact information site-wide.

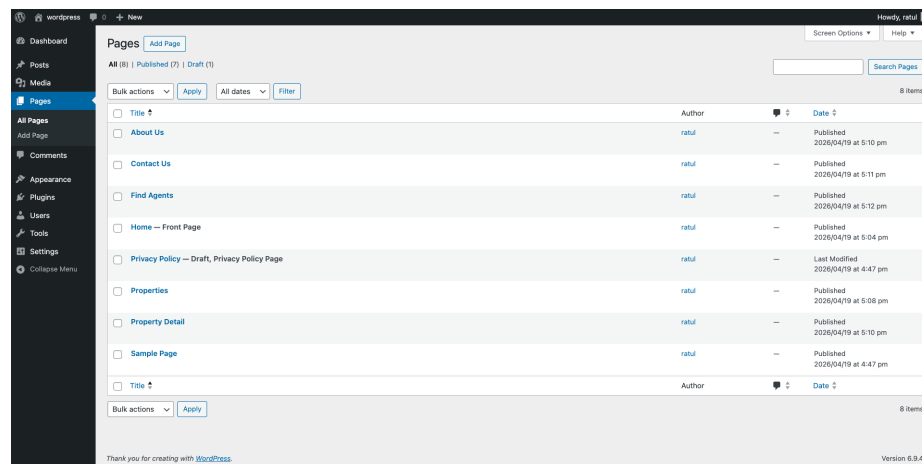


Figure 6.3: WP Admin Pages list with all 7 pages and templates assigned

6.5 Impact of the Internship

This internship provided GrewDev with a custom WordPress theme that is ready for production and tailored to the real estate sector. Since the theme is lightweight and easy to maintain, the agency can deploy it with clients who require better SEO and branding than a typical page-builder site can offer. I also attempted to relieve senior developers of real workload by constructing a solid, modular base. Functionalities

such as the property filter and agent directory are self-contained and do not require intervention every time a change is needed.

I also developed the theme with future work in mind. Phase 2 additions such as custom post types for property listings will not require restructuring the existing PHP templates or CSS files. Using The Loop and standard WordPress template tags throughout the codebase means the theme will remain compatible with future updates to WordPress core. This was in a bid to leave GrewDev with something they can genuinely build on — not merely a completed site, but a foundation that will ease the development process at the next level.

Chapter 7

Conclusions and Recommendations

7.1 Conclusion

After 10 weeks at GrewDev, between February 10 and April 15, 2026, we delivered GrewEstates a handcrafted WordPress theme. The project consisted of seven pages: Home, Properties, Property Detail, About Us, Contact Us, Find Agents and a custom 404 template. The stack consisted of PHP 8.1.11 (modular), Bootstrap 5.3, and Vanilla JavaScript, and five reusable components such as `property-card.php` and `agent-card.php` ensured that the codebase was DRY. I also established conditional loading of assets in `inc/enqueue.php` in such a way that it will only load assets that each page needs, making it significantly lighter than most multipurpose themes.

The most significant change to me was the transition between academic PHP and the real operations of WordPress. My college projects such as the Madina Marble MERN application enabled me to have end-to-end control. WordPress is not that way — you call into the core lifecycle with actions and filters rather than calling items directly. I also needed to adhere to strict coding conventions I was not accustomed to: the need to escape output with `esc_html()` and `esc_attr()`, and prefix `grewestates_` on all functions to prevent collisions. Being supervised by a Project Manager and a Senior WordPress Developer was quite an educative experience of how real client projects work. In retrospect, the internship provided me with a solid foundation in theme architecture, asset management, and responsive UI which I will carry forward into full-stack development in the future.

7.2 Limitations

The GrewEstates theme appears and functions as it should on screen, but there exist actual technical gaps that I would be remiss to ignore. The largest one is that the property listings remain static — the data is not stored in the WordPress database as a Custom Post Type, but rather hardcoded in PHP arrays within the template files. This means there is no admin panel through which a non-technical user can add or update listings without manipulating the code directly. The front-end search box is also responsive and complete, but the backend filtering logic for price ranges and location is yet to be developed.

I also had testing limitations that I was unable to address within the 10-week period. Local testing on WampServer covered only Google Chrome and Microsoft Edge, so I cannot confirm exactly how the theme would perform on Safari or Firefox. Neither did I deploy any automated testing framework — all QA was conducted manually. These gaps are not unexpected given the scope. This phase was always focused on custom theme architecture and styling rather than full database integration or automated pipelines. Those are natural next steps for Phase 2.

7.3 Recommendations

Phase 2, as a way of making GrewEstates fully dynamic, would begin with a proper database-driven content model. The `register_post_type()` function can be used to establish a dedicated CPT, and the Advanced Custom Fields (ACF) plugin would allow non-technical staff to manage price, bedrooms, and area directly through the WordPress dashboard. Subsequently, exposing a WP REST API endpoint for property listings would open up opportunities for future integrations such as React-based frontends or mobile apps built on React Native — a stack I already have experience with.

On the business front, GrewDev may consider adopting WooCommerce to accommodate premium listing charges and agent subscription plans, which aligns with the agency's marketing goals. A caching solution such as W3 Total Cache or WP Super Cache would also be recommended to reduce server response time and improve SEO before going live. It would additionally be worth conducting

a Google PageSpeed audit and an accessibility review addressing WCAG 2.1 AA requirements. These are not features to be added later — as a platform branding itself as a high-quality real estate solution, performance and accessibility must be considered from the outset.

Chapter 8

Acknowledgments

I would like to express my deep and sincere gratitude to my industry supervisor, *Abdullah Ziyad*, Founder and CEO of GrewDev, for giving me the opportunity to conduct this internship and for providing invaluable guidance throughout the entire project. His clear technical direction, vision for quality, and willingness to mentor a junior developer made this experience genuinely transformative. Working under his leadership taught me not only how to write WordPress code but how to think like a professional developer accountable to real client expectations.

I am equally thankful to my academic supervisor, *Suravi Akhter*, Lecturer in the Department of Computer Science and Engineering at the University of Liberal Arts Bangladesh. Her consistent encouragement, timely feedback, and guidance on how to document and reflect on technical work have shaped both this report and my professional outlook.

I am greatly indebted to the honorable teachers of the Department of Computer Science and Engineering at ULAB who taught me during the course of my undergraduate study. Without their rigorous instruction in algorithms, software engineering, web technologies, and system design, I would not have had the conceptual foundation to absorb what this internship demanded of me.

I am extremely thankful to my parents for their unconditional love, their prayers, and their unwavering support through every semester of my degree. Their sacrifices made my education possible. I also thank my friends and classmates, whose companionship and collaborative spirit made the difficult parts of this journey bearable.

Finally, I would like to thank everyone at GrewDev who welcomed me into the team, answered my questions patiently, and gave me the space to learn by doing.

Appendix

Internship Offer Letter

The following document is the official internship offer letter issued by GrewDev – Growth-Driven Web & Marketing Agency, confirming the position of Junior WordPress Developer (Intern) commencing February 10, 2026.

GrewDev – Growth-Driven Web & Marketing Agency

#33 Munshi Bari Road, Zigatola, Dhanmondi – 1209
Dhaka, Bangladesh

Date: February 8, 2026

Subject: Internship Offer Letter – Junior WordPress Developer**To:**

Shaikh Radwan Ahmed Ratul
91 Lalbagh
Dhaka, Bangladesh
Phone: 01715451470
Email: ratulrs29@gmail.com

Dear Mr. Ratul,

We are pleased to offer you the position of **Junior WordPress Developer (Intern)** at **GrewDev – Growth-Driven Web & Marketing Agency**.

After reviewing your background and discussions with you, we believe you have strong potential to grow within our team and contribute meaningfully to our projects. We are excited to welcome you as part of our development team.

1. Internship Position Details

Position Title: Junior WordPress Developer (Intern)
Department: Web Design & Development
Reporting To: Project Manager / Senior WordPress Developer
Internship Type: Unpaid Internship
Start Date: February 10, 2026
Duration: 3 Months (extendable based on performance)

2. Internship Objective

This internship is designed to provide hands-on experience in real-world web development projects. During this period, you will work alongside experienced developers, designers, and project managers to gain practical knowledge in:

- WordPress installation and configuration

- Theme customization (Elementor / Custom Themes)
- Plugin integration and basic customization
- Website speed optimization
- Basic SEO implementation
- WooCommerce setup and product management
- Bug fixing and troubleshooting
- Website security best practices

3. Roles & Responsibilities

As a Junior WordPress Developer Intern, your responsibilities will include:

- Assisting in developing and customizing WordPress websites
- Supporting senior developers with frontend and backend tasks
- Performing website testing and quality assurance checks
- Updating content, pages, and product listings
- Assisting in performance optimization and security checks
- Learning and applying best development practices
- Maintaining proper communication with the project manager

You are expected to demonstrate professionalism, punctuality, willingness to learn, and a proactive attitude.

4. Compensation

This is an **unpaid internship position** intended for skill development, training, and professional growth.

Upon successful completion of the internship period and satisfactory performance, you may be considered for a **paid Junior WordPress Developer position** at GrewDev, subject to availability and performance evaluation.

5. Code of Conduct & Confidentiality

During your internship:

- You must maintain confidentiality of all client data, project information, and internal documents.
- You may not share project files, credentials, or client communication externally.
- You are required to follow GrewDev's internal policies and team guidelines.

Any violation of company policies may result in termination of the internship.

6. Termination Policy

This internship may be terminated:

- By either party with prior notice
- Immediately in case of misconduct, breach of confidentiality, or non-performance

7. Certification

Upon successful completion of the internship period, you will receive an **Internship Completion Certificate** from GrewDev acknowledging your experience and contribution.

Acceptance of Offer

We are confident that this internship will help you strengthen your WordPress development skills and build a solid foundation for your career in web development.

Please confirm your acceptance of this offer by signing below.

We look forward to working with you and seeing you grow with GrewDev.

Warm regards,

Abdullah Ziyad

Founder & CEO

GrewDev – *Growth-Driven Web & Marketing Agency*

info@grewdev.com

I, Shaikh Radwan Ahmed Ratul, accept the position of Junior WordPress Developer (Intern) at GrewDev under the terms and conditions stated above.

Signature: _____

Date: _____

References

Bootstrap. (2024). *Bootstrap 5.3 documentation*. <https://getbootstrap.com/docs/5.3/>.

Google Fonts. (2024). *Inter font family*. <https://fonts.google.com/specimen/Inter>.

WampServer. (2024). *Wampserver official documentation*. <https://www.wampserver.com/>.

WordPress.org. (2024a). *Wordpress codex*. https://codex.wordpress.org/Main_Page.

WordPress.org. (2024b). *Wordpress coding standards*. <https://developer.wordpress.org/coding-standards/wordpress-coding-standards/>.

WordPress.org. (2024c). *Wordpress developer resources — theme handbook*. <https://developer.wordpress.org/themes/>.