**Chapter 4: Development Stages – A Closer Look (with Screenshots)**  
**4.1 Stage 1: Understanding the Requirements**  
The first step in building the Smart CGPA Advisor was figuring out exactly what the system needed to do—and getting that part right was crucial. I started by digging into Caleb University’s official grading policies and the layout of its student results database. Basing everything on actual university documents meant I didn’t have to guess; I could focus on creating something that fit the university’s academic structure perfectly.

During this stage, I also took a close look at how GPA and CGPA are typically calculated in other academic settings and what kind of challenges schools often face when managing student records. As mentioned in Chapter 1, this research was key to making sure the Advisor's calculations would be both accurate and reliable. I explored academic papers, case studies, and other trusted sources to learn what works, what doesn’t, and how to avoid common errors in GPA systems.

To get the project off the ground, I had to first wrap my head around how the grading system at Caleb University actually works. I spent quite a bit of time looking through official documents—grading tables, credit allocations, all of it—to figure out how grades translate into grade points, and how credit units play a role in that process

The deep dives gave me insights that allowed me to anticipate what users would need as well as their pain points. I made a full list of what I wanted. Besides function, I also want performance, security, and look to be non-functional requirements. Caleb University grading rules come with challenges that need to be interpreted in detail for correctness. For example, the college had a specific mark which has to be related to the grade point average of the different programs that need to be mentioned in the application. To clear up any uncertainties, I went to the academic staff who were helpful in doing so. The collaboration was really important in helping us with system requirement, without losing compliance or correctness. I conducted sessions with stakeholders to get their input and confirm the requirements. The university conducted faculty product review sessions for faculty members to share what they liked, didn’t like, and suggestions for improvement. Screenshot

1: [Insert screenshot of Caleb University document or grading policy database schema].

Screenshot 2: [Screenshot of requirement analysis overview]

**4.2 Stage 2. System Design**

I moved into the design phase, after having understood the requirements completely. To show how Smart CGPA Advisor will work within the Caleb University ecosystem, I created detailed architecture diagrams. At this moment, the things that were abstract, needed to be made concrete (as discussed in Chapter 3). The architecture design was focusing on how the web application would retrieve data of student results obtained from the university and process it using various other databases. The areas for input were designed to accommodate important information such as student IDs, course codes, grades, and credit units, as determined by the University’s documentations. I also thought about how students and academic advisors and administrative officers would use the application. Each type would have this relevant paraphernalia available with them. The student and teacher focus were well kept and it was made very simple so users can use it easily. I drew wireframes to see how the screens would look like. And where the buttons, where the input fields, and where information will go. I avoided making the interface too complex but I made sure security was taken into consideration throughout the design. There is protection against misuse of sensitive academic data in school. I made access restrictions according to user roles. Authorized users could modify or view the data. I implemented role-based access controls to ensure that only authorized users could view or modify specific data.

Even though I didn’t get to make a user dashboard, which is what you see in a lot of other applications, I did create wireframes and architecture diagrams to serve as a reference to development and maintain institutional alignment. I also included feedback from the usability testing sessions with the mock-ups where potential users interacted with the designs and provided feedback on their experience. In this phase, the design should be usability oriented but robust enough. To counter this, I continually return to the architecture, getting informal feedback from colleagues to validate assumptions and improve flow. I have also evaluated my peers’ design documents to make sure all bases have been covered in the design and it is designed according to the best UX practices.

Diagram of the system architecture 3: [insert].

**4.3 Stage 3. Implementation.**

Implementation is that stage when the plan meets reality. This happens after the coding of the Smart CGPA Advisor. Built a responsive and accessible interface using familiar frontend technologies such as HTML, CSS, javascript, Bootstrap which functions smoothly on different devices and for different user skill levels. \*This addresses the user-centered values in chapter 2. At the heart of the backend, I developed the components to calculate GPA and CGPA correctly and according to the university's official formulas: GPA Total Grade Points Earned Total Credit Hours (from each semester) CGPA Total Cumulative Grade Points Total Cumulative Credit Hours (for all semesters) The computing engine also accounted for academic situations such as retaken classes, incomplete grades, and many other possible scenarios revealed during previous analysis of the system requirements. With all students to be captured in Caleb University's database architecture, accounting and data fetching were automated and minimized error, because it was all manual previously. Despite some technical challenges, I focused on building clean, well-documented code that would make maintenance and further developments easy, which is consistent with the iterative aspect of developing in chapter 3. I also created error handling to deal with potential errors to avoid crashes to the system, and made sure users would get the correct feedback if they submitted inaccurate data. As I developed the application, I often checked the workings of individual elements to ensure they were performing as intended. I checked components such as GPA and CGPA calculations were working with unit tests, as these were at the core of the application.

I then went ahead to ascertain that, at least, every aspect of the system was reliable and robust before I moved on to the next stage. It was during this stage when I collaborated with other developers. We made it a point to conduct regular code reviews: exchanging great ideas and best practices for their implementation with a culture of improved performance and continuous learning. Such an environment improved the quality of code and made the team more cohesive.

Screenshot 4: [Insert screenshot of GPA/CGPA calculation code snippet and application running with sample data]

Screenshot 5 : [Insert screenshot of data entry interface with validation rules]

**4.4 Stage 4: CGPA Improvement Recommendations**

Stage 4 was the turning point for the Smart CGPA Advisor- from being a mere grade computation tool to being an true academic colleague for students. At this stage, the Advisor began giving personalized recommendations touching on each student's unique educational experience. The system will also know where students are doing great and where they might need a bit more attention after understanding the trend of their current GPA and CGPA. It doesn't just give numbers to the students; it clearly helps them to see their strengths, and softly brings out the areas of concern.

This would suggest courses and electives that their aptitudes and areas for improvement could support. For example, if someone is really good in math but has trouble in humanities, this would advise taking some more math classes or referring to other resources to help strengthen that area.

But that isn't all. The Advisor also provides customized study strategies, ideal study schedules, tutoring options, and material that would make it easier for students to confront some of the harder subjects. Also, within the app, students can create their own academic objectives, such as elevating their GPA by the end of the semester, and the Advisor keeps track of their progress while advising accordingly.

A major part of this stage was the development of a student feedback mechanism from which suggestions could be rated for help in making the system continuously better and more relevant to students' experiences. I worked very closely with academic advisors to ensure the advice does fit tightly within university policy as well as the resources available at the school. Privacy and data security were really foremost because it was valuable to students in that their information is safe and confidential.

Real student data and simulations were put to rigorous testing of the recommendation engine for accuracy, usefulness, and ease of navigation. The tool was tested by students and advisers, who provided invaluable feedback which further refined the tool.

This new incarnation of the Smart CGPA Advisor advanced beyond being merely a calculator. It transformed into a much more personable guide, offering sympathetic advice and encouragement to guide each student through his or her path to success, individualized for his or her needs and goals.

Screenshot 6: [Insert screenshot of the recommendations interface featuring suggested courses, personalized study strategies, and goal tracking.]

**4.5 Stage 5: Deployment and Feedback**

The transition from development to production is often challenging, but the Smart CGPA Advisor was carefully deployed on Caleb University’s web infrastructure with minimal disruption. I coordinated with the university’s IT department to ensure a smooth rollout, addressing any technical concerns that arose during the deployment process.

I conducted a pilot rollout, initially releasing the application to a limited user group. This strategy allowed for controlled observation and quick resolution of any unforeseen issues. Training sessions and comprehensive user manuals were provided to equip users with the confidence and knowledge to fully leverage the Advisor’s capabilities.

Making the application publicly available to the wider university community was an exciting milestone that brought the project’s purpose into full view—empowering students and staff with precise academic information at their fingertips. I felt a sense of pride knowing that the tool I had developed would play a role in supporting students’ academic journeys.

Seeing the application in active use was rewarding, and ongoing communication channels have been set up to gather user experience insights, ensuring the system evolves gracefully with the university’s needs. I encouraged users to share their thoughts and experiences, fostering a collaborative environment where continuous improvement could thrive.

Screenshot 7: [Insert screenshot of live application dashboard]

**4.6 Continuous Improvement and Maintenance**

The cycle of development does not end at deployment. Understanding this, I committed to continuous improvement, embracing regular maintenance, performance monitoring, and iterative updates informed by user feedback and academic policy changes.

Maintaining data security and complying with evolving privacy regulations have been top priorities. The system is regularly audited to detect vulnerabilities and patched accordingly, preserving user trust. I also established a routine for updating the application to incorporate new features and enhancements based on user requests and changing academic requirements.

User suggestions have sparked enhancements such as interface refinements and added reporting functions—demonstrating the value of an adaptable, user-centered approach as highlighted in Chapter 2. I believe that listening to users is key to creating a tool that truly meets their needs.

While adapting to changing academic requirements has posed challenges, scheduled reviews and collaborative planning meetings ensure the system stays aligned with both technical and institutional goals. I remain committed to fostering a culture of continuous improvement, where the Smart CGPA Advisor evolves alongside the university’s academic landscape.

Chapter 4 Summary

This chapter has walked you through each stage of the Smart CGPA Advisor’s development journey, with illustrations signaling key milestones. Rooted deeply in Caleb University’s official policies and data frameworks, this project has yielded a reliable, intuitive, and efficient academic tool.

By moving carefully from requirements analysis through design, coding, testing, deployment, and maintenance—as detailed in Chapter 3—and by embracing continuous feedback and improvement in line with Chapter 2’s user-centric principles, the Advisor now stands ready to support the academic success of Caleb University’s students and staff.

As I reflect on this journey, I am reminded of the importance of collaboration, adaptability, and a user-focused mindset in software development. The Smart CGPA Advisor is not just a tool; it is a testament to the power of technology in enhancing the educational experience.