**PRJ666 – Lessons Learned - TailTales Project**

* Aryan Jitendrakumar Tuwar – Project Manager
* Haroon Ahmed Bajwa
* Vivian Zhonghui Liu
* Mrinaal Nagpal

### **What Went Wrong?**

One of the main challenges we encountered was time management and feature prioritization. Although we made early progress on static components like the Navbar, Footer, and HeroSection, we underestimated the complexity of later features—particularly the appointment booking system and cart functionality. As a result, these core user interaction features were pushed into the final stages of the project, increasing pressure as the deadline approached.

Another key issue was inconsistent task tracking. While development was ongoing, we occasionally struggled to maintain clear visibility on who was working on what. The lack of regular updates or progress logs made coordination more difficult, especially when features like the cart and appointment systems required input from multiple team members. This affected the speed at which interdependent components could be developed and tested.

We also encountered technical issues with cross-browser compatibility and API response times. For example, the Profile page showed inconsistent behavior across different browsers, and our backend endpoints experienced lag under load. These issues required extra debugging time and highlighted the need for earlier performance testing.

### **What Went Well?**

Despite the challenges, several aspects of the TailTales project went very well. The team maintained strong communication and collaboration throughout. Even with scheduling conflicts, everyone was committed, responsive, and willing to adapt to shifting needs. This positive team dynamic allowed us to troubleshoot issues quickly and maintain momentum across different parts of the project.

We were also successful in implementing key technical features, especially the secure authentication system using Firebase. This was a critical milestone that laid the groundwork for user account management and appointment handling. The authentication flow was tightly integrated with profile components and demonstrated strong architectural planning.

Another success was our approach to UI/UX development. Components like TrendingProducts, Contact, and BlogSection received visual polish and usability improvements over time. We prioritized a clean and intuitive user interface, and this helped the site feel more cohesive and professional. Our structured approach to organizing the codebase—with a clear separation of components, styles, and services—also made collaboration smoother and reduced merge conflicts during development.

### **What Could Have Been Better?**

We identified several areas where improvements could be made for future projects:

* **Early Feature Planning**: More time should have been spent mapping out feature complexity early on. Booking systems and shopping cart logic require more time than static components, and pushing them to the end creates unnecessary stress.
* **Task Visibility**: Using a shared task board or project management tool (like Trello or Jira) more consistently would have improved coordination. Having clear assignments and visible progress would have helped prevent overlap and made it easier to identify blockers.
* **Performance Planning**: Earlier testing for performance and compatibility would have caught issues sooner. For example, API latency and browser-specific bugs on the Profile page could have been addressed during mid-stage testing rather than becoming last-minute fixes.
* **Integration Testing**: Although our components were well-structured, integration testing across modules (e.g., login → profile → appointment booking) could have been done earlier and more frequently to avoid surprises in the final stages.

### **Lessons Learned & What We’d Do Differently Next Time**

This project taught us that a strong start doesn’t always guarantee an easy finish—especially when critical features are delayed until the end. In the future, we’d shift our development strategy to *“functionality-first”*—focusing earlier on complex, user-driven features like booking systems, carts, and profile management. These features form the backbone of an interactive application and deserve more time than we originally budgeted.

We also learned the importance of structured communication. Going forward, we’d set up weekly task check-ins and use tools like Jira more proactively to log tasks, bugs, and assignments. This would help prevent confusion about ownership and give everyone a better sense of overall progress.

From a technical perspective, we would prioritize early performance and compatibility testing. Instead of waiting until features are completed, we’d introduce mid-project testing checkpoints to catch issues with API latency, browser compatibility, and component interactions.

Overall, *TailTales* was a fun and rewarding project. Despite the hurdles, we successfully built a functional and engaging pet services website. With clearer planning and earlier focus on key features, we’re confident our next projects will be even smoother and more successful.