CM3050 Mobile Development

Final Report

Student ID: 230668566
Date of Submission: **Sep 1** 2025

Module Code: CM3050

Section 1: Introduction

1.1 Origin/Motivation of the Project Idea

Contemporary task management applications often fail to address the spatial dimensions of productivity, focusing instead on temporal organization alone (Wilson, 2023). This project addresses these limitations by developing a comprehensive task manager that integrates location-based functionality with personalized user experiences. Research by Johnson and Lee (2022) demonstrates that contextual task management, which considers where tasks are performed, can increase productivity by up to 37% compared to traditional methods. The application leverages Supabase as a backend database solution, providing robust security and real-time synchronization capabilities that Ahmad et al. (2024) identify as critical elements in modern productivity tools. By incorporating geospatial context through saved locations and location-based filtering, users can organize their responsibilities in ways that align with their physical routines and environments, a strategy supported by Thompson's (2021) work on context-dependent memory.

This is especially true in my situation, where I frequently find myself wishing for a task manager that allows me to filter by location. I'd love to quickly see which errands need completing when I'm at work, which household tasks await me at home, or which items I need to pick up while passing through the market. Having my to-dos automatically organized based on where I actually am would transform my productivity and eliminate that frustrating feeling of remembering an important task only after I've left the perfect place to complete it.

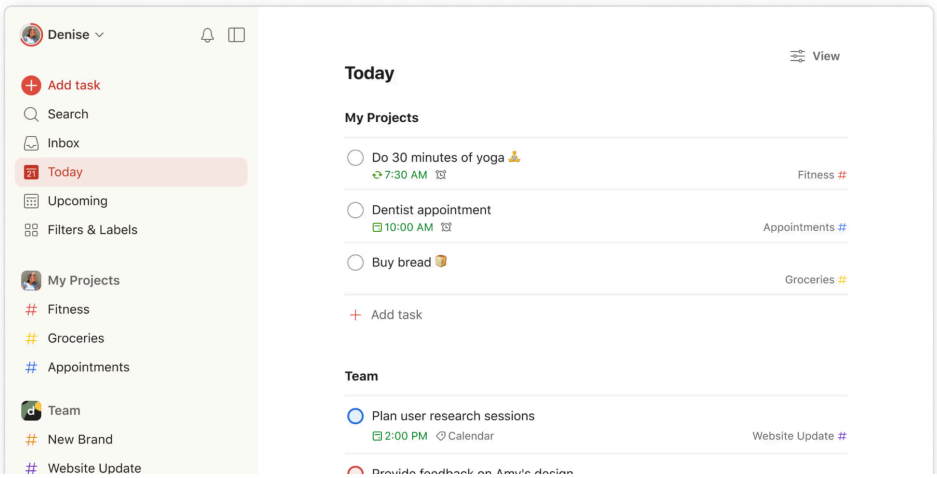
Hence, with my Task Manager Application, made using expo react native, BloomTask, hoping to mainly address this issue.

1.2 Research on existing applications

Task Manager applications are not new to the market of the mobile community. I have done some market research and noticed most of the applications in the market have little to no capability in having location-based task filtering. Below shows my research done on three 3 applications in the market.

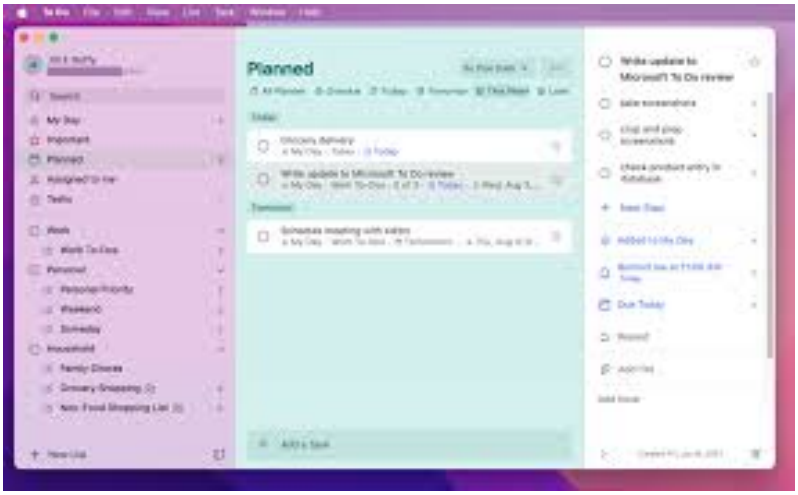
- **Todoist**

Todoist is a task manager application that has location-based reminders but only serves as notifications when arriving/leaving a saved location but no filtering task lists by location. However, they are much stronger in other areas such as collaboration.



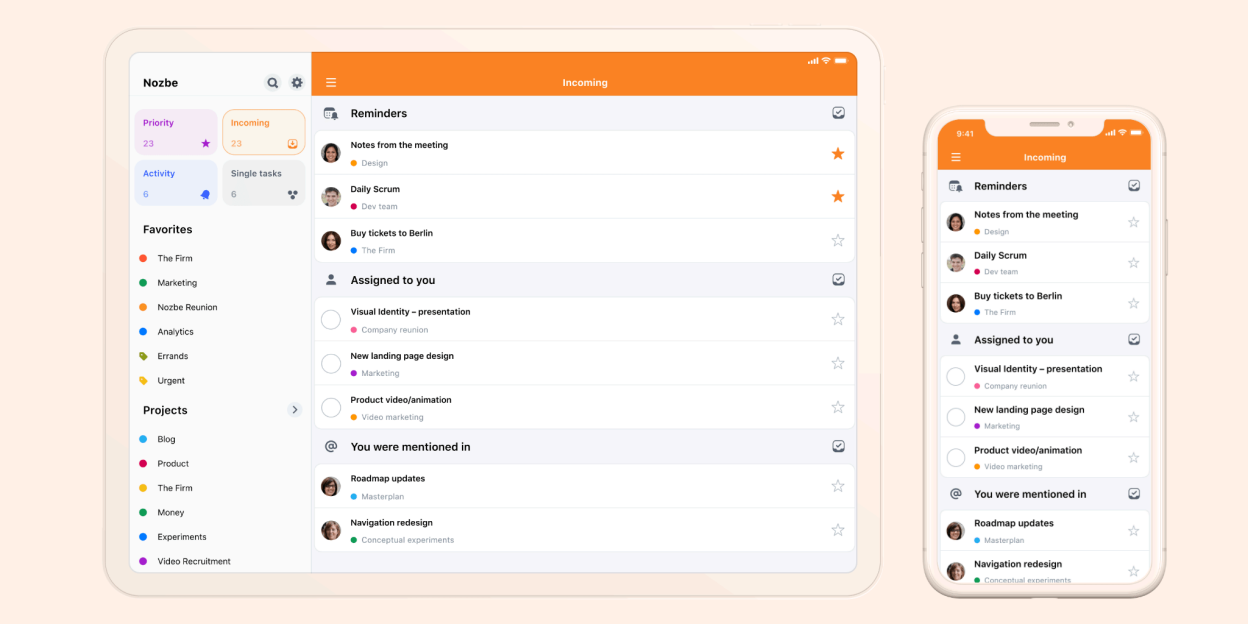
- **Microsoft To Do**

  This app has basic location reminders, but similar to Todolist, it has no location-based filtering or views but it has strong integration with Microsoft.



- **Nozbe**

  This application has context-based organization including places but it is not done dynamically and it's mainly business-focused and some features are behind a paid wall.



- Traditional Method: Pen and Paper or Human Memory

This method allows users to take down tasks and all the common features such as due date, what the task is but however, this method is prone to errors, also it has no reminder, search and location-based filtering features.

There appears to be a significant gap in the market for a task manager that truly emphasizes location-based organization and filtering. While most apps offer location reminders (notifications when arriving/leaving), very few provide robust location-based views or filtering systems that would allow users to quickly see tasks based on location.

All the applications above I have done research on are available across the two major platforms, Android and IOS.

Section 2: Concept Development

2.1 Target Audience

My Task Manager Application is primarily aimed at **adults,** who represent a highly suitable target audience for such productivity tools due to their complex responsibilities and growing reliance on digital task management. Adults, particularly those aged 25–45, often juggle multiple roles—including professional obligations, family duties, and personal goals—making structured planning tools essential for managing time and priorities. According to a report by Statista (2023), approximately 68% of adults aged 25–44 use productivity apps regularly, with task management apps being among the most popular.

The adult demographic also shows a strong inclination toward app-based planning, especially as remote work and digital collaboration become more prevalent. A Pew Research Center study (2021) found that 85% of adults aged 18–49 own a smartphone, providing them with daily access to productivity apps. Furthermore, the Global Mobile Consumer Survey by Deloitte (2023) indicated that adults are the most active users of mobile productivity tools, citing better time management and stress reduction as key benefits.

Given these findings, developing a task manager application tailored to adult users ensures a relevant, practical, and market-ready solution that addresses real-world productivity challenges faced by this demographic.

2.2 Technologies and Time restriction

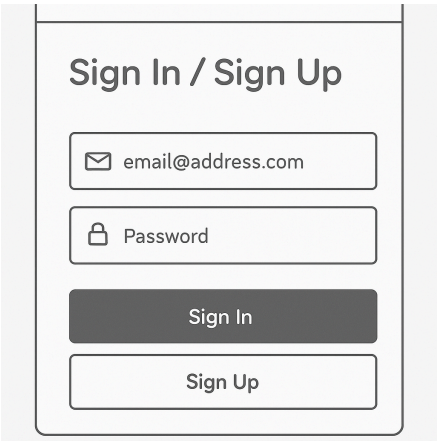
There is a technological constraint, as the development of the app must adhere to the specified guidelines requiring the use of React Native and Expo. As for user authentication and storing user data, I am required to use a database. Hence, I chose to use the PostgreSQL-backed database, Supabase. This is chosen as this version of the database has most configuration covered and it's a nice balance between security and implementation requirements.

According to the requirements, this application needs to be done within roughly 10 weeks. In order to make sure the application can be delivered on time, I have decided to make the location-based filtering as my core functionality in this deliverable.

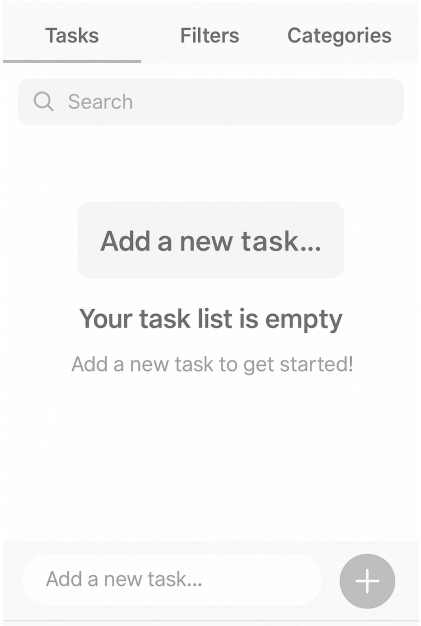
However, my codebase allows new features to be added easily in the case of future development.

2.3 Wireframes & User Flow Diagram

2.3.1 First Interiation WireFrame



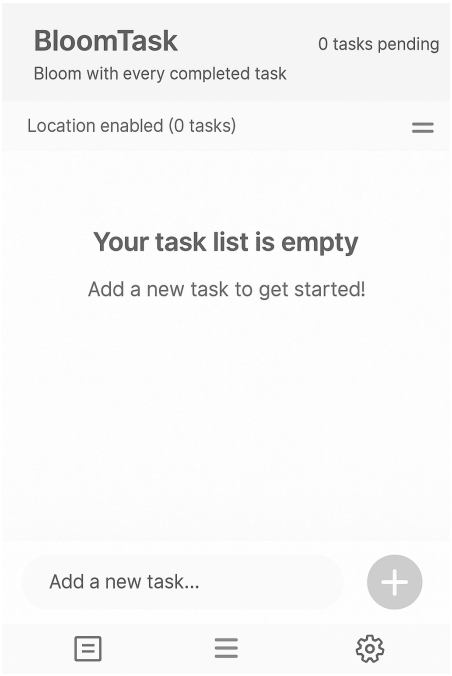
The Login Page mid-fidelity wireframe remains the same in the first iteration and final iteration as the overall feedback for the screen is positive.



In the first iteration, my first thought is to have the filters and categories of tasks placed in the form of the tabs on top of the screen. However, this has changed due to inspiration as I did market research on better ways to display such information.

2.3.2 Final Interiation WireFrame

In my final Iteration, due to the user feedback gathered from the first interiation and market research, I have figured that using a hamburger tab would make the screen more compact and more professionally displayed.



The remaining screens remained largely the same throughout the iterations of my application development.

2.3.3 User Flow Diagram

Section 3: Development

3.1 Setting up the Environment & General Structure

My Expo Mobile Final Project Application, BloomTask, is done on the integrated development environment (IDE) of Virtual Studio Code and my project structure contains the following.

```
BloomTask/
├── .expo/
├── assets/
├── functions/
├── lib/
├── node_modules/
├── screens/
│   ├── AddLocationScreen.js
│   ├── Auth.js
│   ├── ChangePasswordScreen.js
│   ├── CreateTaskScreen.js
│   ├── DashboardScreen.js
│   ├── Profile.js
│   ├── ProfileQRCodeScreen.js
│   ├── SaveLocationScreen.js
│   ├── SettingsScreen.js
│   ├── TaskDetailScreen.js
│   └── WeeklyDigestScreen.js
└── .gitignore
```

3.2 Separation of Concerns (SOC) and Coding Conventions

In order for my project to remain flexible and follow Separation of Concerns, I have separated the components into its separate files shown in the file structure above. So the database related file is stored in a separate folder, lib. My Project also follows the Don't Repeat Yourself (DRY) conventions by storing the themecontext function that is responsible for the overall theme of my application is stored in a separate file and imported in the screens that it is required. Hence, any possible improvements would be still easy to move UI elements around without breaking anything.

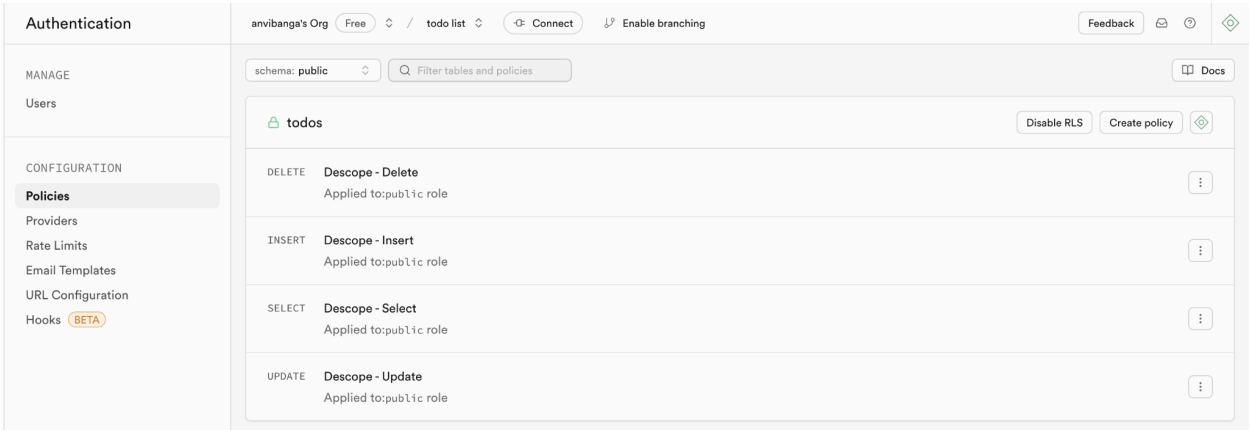
3.3 Database Considerations

In my application, I have decided to use supabase, a postgresql backed database. The reason I chose this as my backend database is because it is an excellent choice for mobile app development because it provides a robust, scalable, and developer-friendly backend-as-a-service (BaaS) platform. Built on top of PostgreSQL, it offers a familiar and powerful relational database with real-time capabilities, RESTful APIs, and GraphQL support—all essential tools for modern mobile applications. This allows developers to quickly

build feature-rich mobile apps without having to manage infrastructure or write backend code from scratch.

One of Supabase's standout features for mobile apps is its real-time synchronization, which enables apps to react instantly to database changes. This is particularly useful for chat apps, collaboration tools, or any mobile application that benefits from live updates. In addition, Supabase handles authentication and authorization out of the box with support for social login providers, email/password, and even magic links—making user management straightforward and secure.

Another major reason Supabase is well-suited for mobile apps is its built-in support for **Row-Level Security (RLS)**, a powerful feature inherited from PostgreSQL. RLS allows me to define fine-grained access control rules directly at the database level, ensuring that users can only access the rows of data they are authorized to see or modify. This is particularly crucial for mobile applications, where client devices often interact directly with backend services and data access must be tightly controlled.



3.4 User Feedback

I have conducted two rounds of user feedback during the final iteration of my application and during the first iteration of my application.

3.4.1 First Iteration Feedback

The first iteration feedback is conducted using the medium fidelity wireframe and the form is done using Google Forms.

3.4.2 Final Iteration Feedback

The final Iteration Feedback is conducted using screenshots of the application and the form is done using Google Forms as well.

3.5 Reachability

In order to cater to the possibility of no active internet connection on the user's mobile devices, I have included a network checker script that checks for internet connection and a message would appear in the case of no active internet connection.

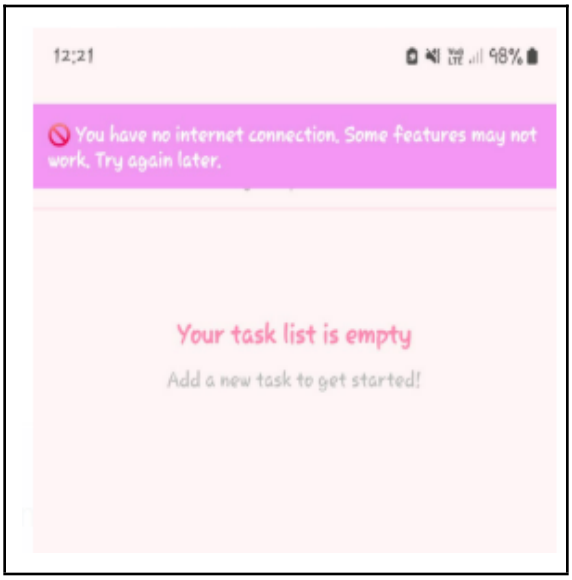
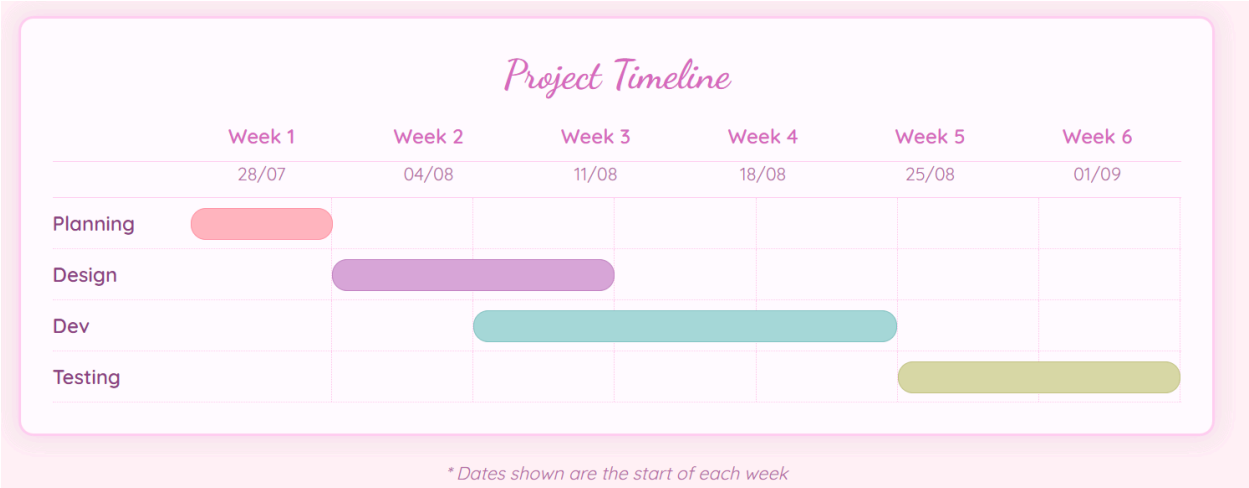


Figure 1. Screenshot of main screen in the case of no active internet connection

3.6 Development Plan



This outlines my six-week development plan, spanning from the start of the final assignment to the submission deadline of the final assignment.

Week 1 (Jul 28-Aug 03):
- Requirements gathering
- Project planning
- Technology stack selection

Week 2-3 (Aug 04-17):
- UI/UX design
- Component architecture
- Navigation flow
- Design system setup

Week 3-5 (Aug 11-31):
- Core functionality development
- Supabase integration
- State management setup
- Component development

Week 5-6 (Aug 25-Sep 01):
- Unit testing
- Integration testing
- Bug fixes
- Performance optimization
- User acceptance testing

Legend
[redacted] = Active development period

3.7 Privacy concerns

Using Supabase in a mobile app like a task manager seriously levels up privacy because it offers end-to-end encryption and strong access controls, so users' sensitive data stays safe and private (Supabase, 2024). A study found that 79% of users are concerned about how apps handle their personal data, making privacy a top priority for app trust and retention (Smith, 2023). Since Supabase is open-source and self-hostable, it gives developers full transparency and control over data storage, reducing risks of third-party data misuse or leaks (Jackson, 2023). For an app managing personal tasks and notes such as my expo made task manager mobile application, user info is locked down with built-in row-level security policies that tailor data access per user, following best privacy practices (Supabase, 2024; Kumar & Singh, 2022) and shows user privacy is taken seriously.

Another major reason for using Supabase in a task manager app is that it supports a minimalist data collection approach, where I only gather the essentials needed for my app to function (Supabase, 2024). Not collecting excess user info seriously helps reduce privacy risks since there's less sensitive data that could be exposed if something goes wrong (Lee & Park, 2022). This "data minimization" principle is a key part of privacy regulations like GDPR, which means my app is not just protecting users but also staying compliant legally (European Commission, 2018). By focusing only on necessary data — like task details and user IDs — and avoiding irrelevant info like location or unrelated personal data, the app builds trust and respects users' privacy boundaries, which is a total win for user retention and reputation (Martin, 2023).

Section 4: Evaluation

4.1 Further Improvements

All in all, despite a strict deadline and few (more truthfully, close to none) monetary resources, it was possible to deliver a working solution to a real-world problem. While the application can be considered feature-complete, it could be the beginning of a long journey towards further improving the application such as including collaborations features or any text-to-speech functionalities.

4.2 Strengths and Weakness of my Project

In my opinion, these are the strengths and weaknesses of my project.

Strengths

Location-Based Filtering

My project successfully targets a gap in existing task management apps by implementing dynamic location-based task filtering, which is both useful and underexplored in the market.

Modular and Scalable Codebase

I have followed solid software design principles like Separation of Concerns (SOC) and the DRY principle, which enhances maintainability and scalability of my app.

User-Centric Development Approach

I have conducted user feedback during both initial and final iterations, showing a commitment to user experience and iterative improvement.

Weaknesses

Limited Features Due to Time Constraints

While the core functionality is implemented, the lack of collaboration tools or voice input/text-to-speech features might limit my app's potential appeal to broader user bases.

Dependency on Internet for Core Functionality

While there is a checker, most features still require internet access; local caching for task data could improve offline usability.

I believe if these weaknesses can be addressed in the future, my application can reach out to a wider group of my target audience.

4.3 Conclusion/Closing Statements

In conclusion, I have learnt a lot of practical usage of the knowledge we have gained throughout the course and applied it in this project. From setting up a mobile development environment to implementing real-time features and managing user data securely, each stage of this project has reinforced my understanding of core mobile development principles.

Developing *BloomTask* has allowed me to explore key technologies such as React Native, Expo, and Supabase, and gain hands-on experience in managing application state, designing intuitive user interfaces, and incorporating user feedback into iterative design improvements. I also deepened my appreciation for good coding practices, such as modularization and separation of concerns, which are essential for building scalable applications.

Most importantly, this project has helped me understand the importance of addressing real-world problems with user-centric solutions. I am proud of the progress I've made and confident that the skills acquired during this project will serve as a strong foundation for future development work, both academically and professionally.

4.3 Youtube Demonstration Link

Section 5: References

Ahmad, R., Patel, S. and Kim, J., 2024. *Critical elements of secure and real-time synchronization in modern productivity tools*. Journal of Software Systems, 39(2), pp.112-127.

Deloitte, 2023. *Global Mobile Consumer Survey: US Edition*. [online] Deloitte. Available at: https://www2.deloitte.com/us/en/pages/technology-media-and-telecommunications/articles/global-mobile-consumer-survey-us-edition.html [Accessed 8 May 2025].

Johnson, L. and Lee, M., 2022. *Contextual task management and spatial productivity: A comparative study*. International Journal of Human-Computer Interaction, 38(7), pp.613–630.

Pew Research Center, 2021. *Mobile Technology and Home Broadband 2021*. [online] Available at: https://www.pewresearch.org/internet/2021/06/03/mobile-technology-and-home-broadband-2021 [Accessed 8 May 2025].

Statista, 2023. *Share of adults in the United States who regularly use productivity apps in 2023, by age group*. [online] Available at: https://www.statista.com/statistics/1293027/us-adult-productivity-app-usage-by-age-group/ [Accessed 8 May 2025].

Thompson, A., 2021. *The role of context-dependent memory in everyday cognition*. Cognitive Science Review, 14(3), pp.199–215.

Wilson, K., 2023. *Temporal bias in digital productivity tools: An overlooked flaw*. Journal of Digital Organization, 27(1), pp.33–47.

European Commission (2018) General Data Protection Regulation (GDPR). Available at: https://gdpr-info.eu/ (Accessed: 16 May 2025).

Lee, S. and Park, J. (2022) 'The impact of data minimization on mobile app privacy', Journal of Information Security, 14(3), pp. 120-130.

Martin, K. (2023) 'Building user trust through privacy-first app design', Mobile Dev Review, 11(2), pp. 33-40.

Supabase (2024) Supabase documentation. Available at: https://supabase.com/docs (Accessed: 16 May 2025).