Conor McNeill
40178155
MSC Individual Software Project
QUB

A dissertation submitted in partial fulfilment
of the requirement for the degree of
Master of Science in
Software Development at
The Queen's University of Belfast

COVER SHEET FOR ASSESSMENT

A signed and completed cover sheet must be attached to each element of coursework submitted for assessment as part of the module. Please complete before submitting your essay.

Work submitted without a completed cover sheet will not be marked.

SUBMISSION TIME IS 12.00 NOON ON FRIDAY 5th SEPTEMBER 2025

Student Name: Conor McNeill	Student Number: 40178155
Module Title: CSC7058: Individual Software Development Project	Module Code: 2241_FYR
Tutor's Name: **Reza Rafiee**	

DECLARATION OF ACADEMIC INTEGRITY

Before signing the declaration below, please note that plagiarism (which includes copying from the work of other students) will be severely penalized. Please check that the submission:

1. Contains full acknowledgement of all secondary sources used (paper-based and electronic)
2. Is clearly presented, word-processed and proof-read
3. Is submitted on, or before, the specified or agreed due date. Late submissions will only be accepted in exceptional circumstances or where a deferment has been granted in advance.

- **I declare that I have read the University regulations on plagiarism, and that the attached submission is my own original work.**
- **No part of it has been submitted for any other assignment, and I have acknowledged in my notes and bibliography all written and electronic sources used.**

Student's signature: Conor McNeill Date of Submission: 5th September 2025

For School office use only *Official School stamp*

ANON CODE	MODULE CODE	Date Received

Acknowledgement

I want to thank my friends, my family and my tutor Reza for being understanding and accommodating in a world where it’s becoming less celebrated.

Abstract

Purpose is a mobile application that delivers curated daily challenges to users, promoting motivation, creativity, and social engagement. Users capture photos or videos completing each task and share them with a close group of trusted friends. The system is built with a modular architecture, featuring an automated challenge generation algorithm, cloud-based media storage, and a responsive user interface. Development followed agile methodologies, emphasizing code quality, maintainability, and scalability. User testing and engagement analysis demonstrate the applications effectiveness in fostering participation and retention, highlighting its potential as a tool for behavioural motivation and community interaction – with limitless capabilities for AI training.

Contents

Chapter One: Understanding the problem

1.1 Introduction

In modern society, daily routines provide individuals with structure, predictability, and efficiency, yet excessive reliance on these routines can reduce social engagement, limit creativity, and suppress intrinsic motivation. Many people, particularly younger generations, spend the limited time they have for themselves engaged in passive activities, such as "doomscrolling"—the compulsive consumption of negative or emotionally charged online content. This behaviour, often facilitated by mobile applications, can exacerbate stress, anxiety, and social isolation, diminishing overall well-being.

Mobile platforms such as TikTok, Instagram Reels, Facebook, Twitter, and Reddit are designed to maximize user engagement through short-form content, notifications, and algorithmic feeds. While these systems are profitable and highly effective at retaining attention, they do not inherently encourage meaningful real-world engagement or promote mental health.

The Purpose App seeks to provide an alternative approach. By offering users a meaningful daily challenge, the application encourages them to capture their experiences via photos or videos and share them with a small, trusted community of friends or family. Unlike attention-driven platforms, the Purpose App prioritizes purposeful engagement, fostering social interaction, presence in the real world, and the creation of lasting personal memories.

This chapter introduces the problem domain, identifies the target audience, evaluates existing solutions, and outlines the requirements and objectives of the project. It also justifies the chosen development strategy, ensuring the application is maintainable, scalable, and user centred.

1.2 Problem Statement

Although structured routines can enhance efficiency and productivity, excessive adherence is linked to negative psychological and social outcomes, including decreased intrinsic motivation, limited creativity, and weakened social engagement[1]. These effects are compounded by the phenomenon of "doomscrolling," wherein individuals consume large amounts of negative or emotionally charged content online. This behaviour has been shown to increase stress, anxiety, and a sense of social isolation, particularly among younger populations[2].

[1] Robillard, R., et al., The rhythm of your life: The effects of daily routine on mental health, Graduate Nursing Journal, 11(1), 2018

[2] Harvard Health Publishing, Doomscrolling dangers, Harvard Health, 2024, https://www.health.harvard.edu/mind-and-mood/doomscrolling-dangers.

Current mobile applications often attempt to address engagement or entertainment needs, but they primarily exploit attention-maximizing mechanisms that prioritize screen time over meaningful user experiences. Platforms such as TikTok, Instagram Reels, and Twitter encourage passive content consumption rather than promoting proactive engagement, reflection, or real-world interaction. Consequently, there is a clear gap in digital solutions that support purposeful behaviour while mitigating the negative effects of excessive routine and passive social media consumption.

The Purpose App addresses this gap by providing users with structured, meaningful daily challenges. These challenges are designed to encourage real-world activity, creative expression, and social interaction within a small, trusted community. By capturing and optionally sharing these experiences through multimedia, users are empowered to balance routine with purposeful engagement, fostering motivation, presence, and personal reflection.

This dissertation documents the design, development, and evaluation of the Purpose App, emphasizing maintainable, scalable, and user-centred practices in software development.

1.3 Target Audience

The primary audience for the Purpose App comprises individuals aged 18–50 who are seeking motivation, personal growth, or creative engagement in their daily routines. This demographic typically includes university students, early to late career professionals, and hobbyists who actively use mobile devices and social platforms for self-expression, social interaction, and information consumption. Users in this group often experience challenges related to maintaining a balanced routine, fostering creativity, and avoiding passive engagement habits such as doomscrolling.

These individuals are likely to be motivated by structured yet flexible interventions that encourage them to engage in purposeful activities while providing opportunities for reflection, documentation, and social sharing. They value personal development, creativity, and social connectedness, and are comfortable navigating app-based interfaces for tracking and sharing experiences. The app is designed to support both solitary reflection and small-scale social interaction, enabling users to maintain meaningful engagement with their immediate carefully chosen community.

Example User Personas:

Persona 1: Sarah, 22

- Profile: University student studying Media and Communications.
- Goals: Seeking inspiration to try new activities, document experiences creatively, and share achievements with her friends.
- Challenges: Limited motivation due to academic stress and tendency to spend free time on passive social media consumption.
- Needs from App: Structured daily prompts that encourage exploration, creative expression, and meaningful social sharing with a trusted circle.

Persona 2: Mark, 28

- Profile: Early-career professional working in finance.
- Goals: Wants to integrate small creative or reflective activities into his busy work schedule to maintain balance and reduce stress.
- Challenges: Time-constrained lifestyle, difficulty maintaining social interactions outside work, and susceptibility to unproductive phone use.
- Needs from App: Quick, actionable daily challenges that are easy to complete, promote personal growth, and can be shared selectively with close friends or family.

Persona 3: Aiden, 47

- Profile: Law firm executive.
- Goals: Interested in friendly competition with peers.
- Challenges: Easily distracted by online content, overworked, and seeks motivation to engage in offline activities.
- Needs from App: Engaging, gamified challenges with options for social interaction, creative documentation, and reflection.

By clearly defining these target audiences and personas, the Purpose App ensures that both the design and functionality cater to users' intrinsic motivations, balancing the benefits of structured routine with opportunities for meaningful engagement, creativity, and social connection.

1.4 Existing Solutions and Gap Analysis

Several applications address aspects of daily motivation or social sharing, yet none combine all desired functionalities effectively. The table below summarises some existing solutions and their limitations:

Application	Daily Challenge	Media Capture	Social Sharing	Notes on Gaps
Habitica	✓	×	✓	Focuses on gamification, not media-based
Daylio	✓	×	×	Tracks moods but offers no sharing or media capture
TikTok	×	✓	✓	Provides social sharing but lacks structured daily purpose

Gap analysis: Existing applications either lack integrated media capture combined with purpose-driven daily challenges or fail to maintain engagement through structured interaction. The Purpose App addresses these gaps by combining challenge generation, multimedia capture, and social community features, offering users both motivation and meaningful engagement.

1.5 Requirements Elicitation

The Purpose App must provide both functional and non-functional capabilities to support meaningful daily engagement.

Functional requirements:

- Generate daily challenges tailored to demographic.
- Capture photo and video evidence of completed challenges.
- Enable sharing within the app and optionally on external social media.
- Support user authentication and profile management.
- Provide notifications and reminders for daily challenges.

Non-functional requirements:

- Scalable backend with cloud storage to handle media content.
- Responsive and intuitive user interface to ensure accessibility and ease of use.
- Secure handling of user data, adhering to GDPR.

Requirements were gathered through literature review on habit-forming apps, analysis of existing solutions, and hypothetical user personas to define key functionalities and usability expectations.

1.6 Project Objectives

The project aims to deliver a fully functional mobile application that provides structured, purposeful engagement for users. The primary objectives are:

- Develop core app features including challenge generation, media capture, and selective sharing.
- Ensure a smooth, user-friendly interface for completing challenges and recording evidence.
- Implement social engagement functions such as sharing, liking, and commenting within a trusted community, (however at this stage it is unknown if public social engagement will be encouraged on this application as its cons might outweigh the pros).
- Design a scalable and maintainable backend architecture capable of handling media uploads and user data efficiently.
- Conduct testing to validate usability, performance, and reliability of the application.

A stretch goal includes exploring AI-based training and using the media received from users.

1.7 Development Strategy

The Purpose App will be developed using an iterative approach, following Agile principles to enable incremental delivery and continuous testing of features.

Methodology: Iterative development with short review cycles to refine functionality, gather feedback, and address technical challenges early in the process.

Tools and technologies:

- Frontend: Swift, chosen to prioritise native iOS development and ensure seamless integration with Apple's ecosystem.
- Backend: Node.js with Express, running locally via server.js on the terminal for API handling and communication between client and server.
- Database: MySQL, hosted locally through MAMP to manage structured relational data such as user accounts, prompt records, and media references.
- Cloud storage: Initially managed locally, with potential for future migration to cloud services (e.g., AWS or Google Cloud) as scalability demands grow.

Justification: Swift ensures optimal performance and user experience on iOS devices, the initial target platform. Node.js provides flexibility and scalability for backend services, while MySQL offers reliable relational data management. Local development using MAMP and Node.js allows efficient prototyping and iteration, with scope to transition to cloud-based infrastructure if needed.

Timeline overview: High-level milestones include requirements finalisation, UI/UX design in Swift, backend setup with Node.js, database integration via MySQL, feature implementation, system integration, testing, and initial deployment. Development is expected to follow a 12-week schedule.

Risk management: Potential risks include media upload issues, notification failures, and constraints from initially focusing on a single platform. Mitigation strategies include rigorous testing of upload and notification functions, maintaining modular code for easier future cross-platform expansion, and ensuring varied, engaging challenges to support user retention.

Chapter Two: User Interface Design

2.1 Overview of UI Design

The user interface (UI) of the Purpose App has been designed to prioritise simplicity, clarity, and user engagement. Since the application targets individuals seeking purposeful daily challenges while avoiding the overwhelming nature of existing social platforms, the interface avoids excessive visual clutter or dark patterns intended to maximise screen time. Instead, the design philosophy centres on clean layouts, minimal colour palettes, and intuitive navigation flows that support efficient task completion.

The app follows a familiar mobile design structure, including a bottom navigation bar, clear iconography, and card-based layouts for challenge prompts and media sharing. This consistency aligns with established iOS design guidelines, reducing the learning curve for first-time users. Furthermore, the design emphasises balance between personal reflection (private capture of experiences) and community interaction (sharing with trusted connections).

2.2 UI Design Mock-ups + Wireframes

The user interface was first developed through sketches before being refined into higher-fidelity mock-ups using Photoshop. These mock-ups were then tested with a small group of representative users, whose feedback informed subsequent iterations of the design.

The following core screens were produced:

- Home screen: Displays the daily challenge prominently at the top, with a central capture button and supporting navigation.
- Capture screen: Integrates the iOS camera interface, providing users with an uncluttered view and a clear shutter button for photo or video capture.
- Feed screen: A small scrollable feed displaying challenge completions shared by trusted contacts, each presented as a card with options for liking and commenting.
- Profile screen: Displays user information, progress statistics, and a record of completed challenges, offering opportunities for reflection.

Figures in the appendices include:

- Figure 1.1: Wireframe of the Capture Screen with upload and daily prompt.
- Figure 1.2: Wireframe of the Feed Screen displaying a limited set of shared posts.
- Figure 1.3: Wireframe of the Profile Screen showing user details and challenge history.

2.3 UI Design Justification and Iteration

The design of the Purpose App evolved through several iterations. Initial sketches were critiqued in informal feedback sessions with friends and professionals, representing the app's intended audience. The following refinements were made as a result:

- Relocation of the capture button: Early designs placed this in the top navigation bar, but users found this unintuitive. It was moved to a bottom floating button for ease of access. The other main screens are accessed via main buttons that float at the bottom of the screen also for further ease of access.
- Colour scheme: Darker tones used in the first prototypes were found to convey a negative or overly serious tone. A lighter, more neutral palette was adopted to reinforce positivity and encourage daily engagement.
- Simplified navigation: Early versions featured deeper menu hierarchies for settings and progress tracking. These were streamlined into single-level menus to reduce friction and improve usability.

The decision to align with iOS Human Interface Guidelines ensures that the app feels familiar to Apple users while retaining its unique identity through branding. This balance of familiarity and originality is crucial for early adoption.

2.4 Accessibility and Usability Considerations

Accessibility and inclusivity were treated as core requirements during the design process. The following measures were implemented:

- Typography and contrast: Clear, legible fonts and high-contrast colour combinations were selected to enhance readability.
- Touch target sizes: Buttons and interactive elements meet Apple's minimum target size recommendations to support users with motor impairments.
- Colour independence: Success states (e.g., challenge completion) are conveyed not only through colour but also with icons, ensuring accessibility for colour-blind users.
- Navigation simplicity: Core functionality is accessible within two taps, avoiding deep navigation structures that may cause confusion.

Usability testing using clickable prototypes highlighted the importance of quick and efficient user flows. For example, unnecessary confirmation screens for media capture were removed to streamline task completion. These changes ensure that the app remains practical for busy individuals while still supporting its goal of encouraging reflection and meaningful engagement.

Chapter Three: System Architecture and Design

3.1 System Architecture Overview

The Purpose App is designed using a client–server model that separates presentation, application logic, and data management. This structure ensures that the system remains modular, scalable, and maintainable, while supporting the user-centred goals of the project. The decision to focus development on iOS devices was informed by the target audience, who are typically active smartphone users, and by the need to prioritise stability and polish in a single environment before considering cross-platform deployment.

The client side, implemented natively in Swift, provides a responsive and seamless user experience in line with Apple's Human Interface Guidelines. The backend, developed in Node.js, acts as the intermediary between the user interface and the data layer, handling authentication, challenge generation, media metadata processing, and user communication. A MySQL database, hosted locally in MAMP during development, provides reliable relational storage of structured data such as users, challenges, and references to captured media. Media files themselves are stored locally, with the architecture deliberately designed to allow straightforward migration to cloud storage solutions in later iterations.

The architecture is divided into three logical layers:

- **Presentation Layer**: The iOS client built in Swift provides the graphical interface, manages navigation, and facilitates user interaction with the system. This layer is responsible for displaying daily challenges, capturing media, and transmitting data securely to the backend.
- **Application Layer**: The Node.js backend acts as the application server, providing RESTful API endpoints for user authentication, retrieval of daily challenges, submission of media metadata, and access to community feeds. It processes requests from the client, enforces validation rules, and manages data exchanges with the database.
- **Data Layer**: The MySQL database stores persistent, structured information such as user credentials, challenge definitions, and references to uploaded media. During early development, media files themselves are stored on the local machine, but this is encapsulated in a way that allows migration to cloud-hosted solutions such as AWS S3 or Google Cloud Storage without substantial redesign.

This layered separation enables independent evolution of each component. For instance, the backend could be scaled by deploying the Node.js server on a remote host while the database is migrated to a managed service, without requiring significant changes to the iOS client

3.2 System Components

The major system components and their responsibilities are as follows:

- **Frontend (Swift iOS App)**:
    - Implements the graphical user interface and navigational flows.
    - Captures photos and videos via native iOS camera APIs.

    - Handles local data caching for offline use.
    - Communicates with the backend via RESTful HTTP requests.
- **Backend (Node.js with Express)**:
    - Provides a modular, event-driven server environment.
    - Hosts RESTful API endpoints for login, registration, challenge retrieval, and metadata submission.
    - Performs validation and basic business logic (e.g., daily challenge distribution).
    - Coordinates with the MySQL database for persistent data storage.
- **Database (MySQL via MAMP)**:
    - Manages structured relational data.
    - Provides referential integrity for user accounts and challenge logs.
    - Stores metadata (e.g., file paths, timestamps, and ownership) for captured media.
- **Storage**:
    - Media files are initially stored locally for development and prototyping.
    - Encapsulation of storage logic ensures a smooth transition to cloud storage in future releases.

This division of responsibilities ensures each component remains cohesive and independently testable.

3.3 Data flow

The data flow between components operates as follows:

1. **Challenge Retrieval**:
    - The user opens the app and requests the daily challenge.
    - The iOS client sends a request to the Node.js backend.
    - The backend queries the MySQL database and returns the appropriate challenge to the client.
2. **Media Capture and Submission**:
    - The user captures a photo or video to evidence completion of the challenge.
    - The client submits metadata (file path, timestamp, and user ID) to the backend.
    - The backend validates the data, stores the metadata in MySQL, and saves the media file in local storage.
3. **Community Feed**:
    - Trusted contacts request updates through the feed screen.
    - The backend retrieves relevant challenge completions from the database and returns metadata for rendering.
    - Media files are displayed via stored file paths or links.

This linear, event-driven process ensures smooth operation while maintaining a clear separation of concerns.

3.4 Design Justification

The architecture was chosen to reflect the project's aims of simplicity, extensibility, and practicality within development constraints. Key justifications include:

- **Swift Frontend**: Using Swift ensures a highly responsive native iOS experience, offering smooth camera integration and alignment with platform standards.
- **Node.js Backend**: Node.js provides a lightweight, asynchronous environment well-suited for handling multiple concurrent client requests. Its modularity also facilitates future scalability.
- **MySQL Database**: Relational databases are appropriate for structured data such as users and challenges, offering referential integrity and mature query capabilities.
- **Local-first Deployment**: Developing locally with MAMP and local storage enables rapid iteration and debugging. The architecture remains migration-ready for deployment in a cloud-hosted production environment.

The modular separation also ensures that each component can be independently tested and replaced if future development priorities shift.

3.5 Risks and Mitigation

The following risks were identified in relation to the system design:

- **Scalability Constraints**: Local MySQL and storage will not scale to large user bases.
  - *Mitigation*: Plan for cloud migration (AWS, Google Cloud) once the prototype matures.
- **Reliability of Notifications**: Push notifications via Node.js may fail, undermining user engagement.
  - *Mitigation*: Employ extensive testing with fallback reminders (e.g., scheduled local notifications).
- **Media Upload Failures**: Large file uploads could disrupt the user experience.
  - *Mitigation*: Implement size restrictions and compress media before submission.
- **User Retention**: Users may disengage after initial novelty fades.
  - *Mitigation*: Offer varied challenge types and explore AI-driven personalisation in future releases.

Chapter Four: Testing

4.1 Overview

Testing is a critical stage in the software development lifecycle, ensuring that the resulting system functions as intended, meets user requirements, and maintains reliability under realistic usage conditions. For the Purpose App, testing was not treated as an afterthought but was integrated throughout the development process. This approach followed the principles of test-driven development (TDD) in parts of the backend and iterative testing in the iOS frontend, ensuring continuous validation of features as they were implemented.

The testing strategy adopted for this project was intentionally multi-layered. Different testing methods were applied depending on the risk profile of each component. The backend APIs, for example, were considered high risk due to their role in authentication, data integrity, and challenge distribution; these were validated extensively using automated unit and integration tests. The frontend user interface, while less technically complex, required usability-focused evaluation to ensure that it aligned with user expectations and encouraged purposeful behaviour. Meanwhile, performance and stress testing were used to assess how the system behaved under load, with particular focus on media uploads and notifications.

This chapter details the rationale for the chosen testing approaches, describes the methods applied, outlines the areas of greatest risk, and provides evidence that the system was rigorously validated.

4.2 Priority Areas for Testing

Although every part of the application was subject to some form of validation, certain areas carried greater risk of introducing bugs or reducing user satisfaction. These areas were prioritised accordingly:

4.2.1 Media Capture and Uploading Content

The process of capturing photos and videos on the iOS client, compressing them, and transmitting metadata to the backend was considered one of the most error-prone aspects of the system. Risks included:

- File size limitations leading to failed uploads.
- Inconsistent behaviour across different iPhone models.
- Network interruptions causing data loss or corruption.

Since media capture lies at the heart of the app's purpose-driven concept, errors here would have undermined the system's value. Extensive functional and performance testing was therefore applied.

4.2.2 Backend Integration

The Node.js backend acted as the “glue” connecting the iOS client with the MySQL database. Bugs at this layer could manifest as authentication failures, incorrect challenge retrieval, or incomplete metadata storage. Furthermore, as backend endpoints are likely to evolve in future iterations, achieving robust test coverage was critical. Scalability was also a priority concern.

4.2.3 Notifications

Daily notifications are central to encouraging engagement. Incorrect scheduling or failures in delivery would significantly reduce user retention. Testing was therefore required to validate both the logic of notification generation and their reliability in different app states (foreground, background, or inactive).

4.2.4 Authentication and Profiles

User authentication was a key area for security risks. Incorrect handling of sessions or tokens could lead to unauthorised access or data breaches. The login, registration, and profile management features were tested extensively with both valid and invalid inputs.

4.3 Testing Methodology

To address these risks, a hybrid strategy was employed. Automated testing ensured repeatability and high coverage for backend logic, while manual and usability testing validated end-to-end flows and user experience.

4.3.1 Unit Testing

Unit tests were written using Jest for backend components and XCTest for selected iOS modules.

- Backend Unit Tests: Focused on small, isolated functions such as password hashing, token generation, and database query responses. For example, one test confirmed that invalid login attempts returned an unauthorized status. Another validated that challenge retrieval endpoints always returned a JSON object with the correct fields.

- Frontend Unit Tests: Limited but targeted, verifying navigation flows and error handling within the Swift app. XCTest was used to simulate button presses and ensure that expected screens appeared.

By isolating small functions, these tests reduced the likelihood of subtle defects propagating into larger system failures that would plague the system in its later development. The approach taken allowed for small testing of almost every important backend and frontend function as it was being developed to ensure its proper function.

4.3.2 Integration Testing

Integration tests validated the flow of data across system boundaries. Examples included:

- Submitting a new user registration form in the iOS client, confirming that the backend API returned a valid token and that a new record appeared in the MySQL database.

- Capturing a photo, uploading it, and checking that metadata (filename, user ID, timestamp) was inserted correctly into the database.

Integration testing was essential to ensure that the modular design worked cohesively in practice.

4.3.3 End-to-End Testing

Manual end-to-end (E2E) testing simulated realistic user journeys, such as:

1. A new user registers, logs in, and retrieves their first daily prompt.
2. A user captures and uploads a video, checks their profile to confirm storage, and later views the submission in the community feed.
3. A returning user receives a daily notification, opens the app, and successfully completes the task.

These tests were carried out repeatedly during development, often after significant feature additions. They provided confidence that the overall user workflow remained stable.

4.3.4 Performance Testing

Performance testing was conducted to assess system behaviour under load and to evaluate media upload times. Stress tests on the backend simulated up to 50 concurrent requests (as with the application in its early use would hold around this amount of members with friends and family using the application first as a sort of BETA test before full publication and by then cloud data will be being used), measuring latency and throughput.

Although resource constraints prevented testing at scale, results indicated that the system could handle prototype usage without significant degradation.

On the client side, upload times were recorded for different file sizes. Results confirmed that compression was necessary for larger video files, and an upper limit of 50 MB was enforced.

It is known that for a more reliable robust performance testing that the server would need to be moved from localhost client to a cloud system that can maintain and perform at industry level and therefore the application would need to be moved to a third part data storage system to allow for greater upload speeds and higher MB file storage.

4.3.5 Usability Testing

Usability testing involved ten participants from the target demographic (ages 18–50). They were asked to perform tasks such as completing a prompt, uploading the media, and sharing content. Observations included:

- Users instinctively located the capture button when completing challenges.
- Some confusion arose from error messages during failed uploads, prompting revisions to wording.
- Notifications were generally effective, but participants recommended varying the wording to avoid repetition or for it to become like the application “BeReal” which became a notification many had to turn off.

This feedback directly informed interface refinements and notification improvements.

4.4 Test Case Design and Coverage

Test cases were designed using systematic techniques such as equivalence partitioning and boundary value analysis:

- Authentication: Valid login, invalid username, invalid password, expired token.
- Media Upload: Image <1MB, video = 50MB (limit), unsupported file type (.gif).
- Notification: App open, app in background, device locked.

By covering both expected and edge conditions, the test suite reduced the likelihood of defects arising in real-world usage.

4.5 Evidence of Testing

The following artefacts were produced to support the testing process, (can be viewed in the appendix of this document)

- Automated Test Scripts: Jest test files for backend APIs; XCTest files for navigation logic.
- Manual Test Plans: Documented scenarios with expected results and actual outcomes.
- Performance Logs: Output from load tests and upload timing benchmarks.
- Usability Reports: Summaries of user sessions, including feedback and suggested changes.

These artefacts have been included in the project deliverables and are reproducible for future developers.

Chapter Five: Evaluation and Conclusion

5.1 Evaluation Against Objectives

The Purpose App was developed to address a clearly defined problem: the lack of mobile applications that promote purposeful daily engagement without resorting to exploitative attention-retention mechanisms. The project sought to deliver a system capable of generating daily challenges, enabling photo and video capture, supporting selective sharing within a trusted network, and encouraging reflective engagement.

Measured against the functional and non-functional requirements outlined in Chapter 1, the app can be considered a partial success. The core functional requirements—daily challenge generation, multimedia capture, authentication, and notifications—were all implemented and tested successfully, as evidenced by the test results presented in Chapter 4. Furthermore, the app provides an intuitive user interface consistent with the design principles set out in Chapter 2. However, some features, such as broader social sharing and long-term user engagement metrics, remain underdeveloped due to time constraints.

From a non-functional perspective, the project achieved satisfactory results. The app runs smoothly on iOS devices, with performance testing demonstrating acceptable upload times and responsiveness under moderate load. Nevertheless, scalability is currently limited; the use of a locally hosted Node.js backend and MySQL database via MAMP restricts the system's ability to handle large volumes of concurrent users. This shortcoming reflects a deliberate decision to prioritise prototyping over deployment readiness.

Overall, the project succeeded in creating a functioning prototype that validates the core concept. It fell short, however, in achieving the stretch objective of incorporating personalised challenges through adaptive algorithms.

5.2 Evaluation of Development Environment

The choice of Swift for frontend development proved advantageous, as it allowed for seamless integration with iOS features such as notifications, camera access, and local storage. Swift also provided a mature testing framework (XCTest), which facilitated structured unit testing. The drawback of this choice is its exclusivity: the current app is locked to the Apple ecosystem, and the absence of a cross-platform framework such as Flutter or React Native limits the potential user base.

On the backend, Node.js with Express offered a straightforward and lightweight environment for handling requests and managing authentication. However, relying on a locally hosted MAMP MySQL server was suboptimal for simulating real-world conditions. While sufficient for prototyping, it restricted scalability and introduced constraints in testing distributed load scenarios. Future work would benefit from deploying the backend to a cloud environment such as AWS or Firebase to ensure realistic performance evaluation.

The development environment, therefore, was appropriate for building and testing a proof-of-concept system, but not for simulating full production conditions.

5.3 Reflection on Project Process

The most successful aspect of the project was the **user-centred design approach**. By defining clear personas and iterating through wireframes and mock-ups, the interface was designed in alignment with user expectations. This resulted in positive usability testing outcomes, where users reported that the system was both intuitive and engaging.

Another strength was the **implementation of multimedia handling**, which was integrated effectively into the iOS ecosystem. The ability to capture and store images and videos, linked directly to daily challenges, was central to achieving the project's objectives.

The least successful aspects were related to **scalability and long-term engagement**. While the backend functioned reliably for small-scale use, it was not capable of supporting larger user bases or ensuring resilience under heavy loads. Furthermore, the app does not yet include features to maintain user motivation over extended periods, such as streak tracking, challenge diversity algorithms, or adaptive personalisation.

In terms of project management, the iterative approach worked well for incorporating feedback and testing features incrementally. However, time constraints limited the breadth of testing and forced compromises, particularly around advanced functionality.

5.4 Future Work

There are several directions for extending and strengthening the Purpose App:

1. Cross-platform support: Re-implementing the frontend in Flutter or React Native would broaden accessibility beyond iOS devices, increasing adoption potential.
2. Cloud deployment: Migrating the backend and database to a cloud-based service would improve scalability, resilience, and performance under real-world usage conditions.
3. Challenge personalisation: Implementing adaptive algorithms (e.g. machine learning based on user preferences and engagement history) could enhance motivation and long-term use.
4. Gamification elements: Adding features such as streaks, points, or badges may provide extrinsic motivation while maintaining the app's ethos of meaningful engagement.
5. Expanded usability testing: Conducting larger-scale trials across diverse demographics would provide more robust evidence of effectiveness and highlight areas for further improvement.
6. Privacy and security auditing: A more rigorous review of data handling practices, especially if the app were to move towards deployment, would ensure compliance with GDPR and best practices in mobile security.
7. Limiting User Followings: A way to limit or improve how users add one another, perhaps a question to answer before you can follow the user, the goal is to not have the application become like an overgrown Facebook or Instagram where users are too scared to post as they follow 500+ people from their family to work colleagues to people they met six years ago on holiday, we believe this leads to an engagement fear for posting and would like to avoid this.

5.4.2 Artificial Intelligence

Looking ahead, artificial intelligence (AI) and machine learning (ML) present considerable opportunities to extend the Purpose App's functionality. One potential application is the use of user-generated media to improve challenge personalisation. For instance, if the app issues a prompt such as "take a picture of a flower," the resulting non-copyrighted images could contribute to datasets for image recognition or generative AI models. In turn, the system could generate more adaptive challenges that respond to users' preferences, behaviours, and engagement history.

Potential AI-driven enhancements include:

- Activity-based adaptation: If users frequently documents outdoor activities, the app could recommend more nature-based or physical prompts. Conversely, users with a preference for creative indoor tasks might receive art, journaling, or reflection-based prompts.

- Temporal and seasonal relevance: AI could suggest challenges aligned with local time, weather, or seasons (e.g., encouraging photography of autumn leaves or a morning mindfulness task).

- Well-being indicators: By analysing engagement patterns, the app could detect reduced activity and respond with lower-effort, motivational challenges to help re-establish routine without overwhelming the user.

- Community dynamics: AI could identify common themes across a user's trusted network and propose collaborative or complementary challenges to strengthen social connectedness.

However, these opportunities introduce critical ethical considerations. Any use of user-contributed media for training datasets must be transparent, voluntary, and compliant with data protection regulations such as GDPR. Clear consent mechanisms, anonymisation of metadata, and options to opt out are essential to maintain user trust. Furthermore, robust safeguards are needed to prevent misuse of collected data and to avoid reinforcing harmful biases within AI models.

If implemented responsibly, AI-driven personalisation could significantly enhance long-term engagement by tailoring prompts to individual motivations while preserving the app's ethos of purposeful and meaningful interaction. By embedding ethical practices from the outset, the Purpose App can demonstrate how AI may be leveraged not only for technical advancement but also for promoting healthier, more intentional digital behaviours.

Finally, moving toward more individualised prompts would fundamentally alter the app's dynamic. While the original concept encourages a shared experience with all users completing the same challenge, personalisation could increase engagement by tailoring prompts to individual preferences and behaviours. This trade-off between collective participation and personalised interaction is an important consideration for future development, requiring careful balancing to maintain both social cohesion and meaningful user engagement.

5.5 Conclusion

The Purpose App demonstrates the feasibility of designing mobile applications that move beyond passive attention capture toward intentional, purposeful engagement. By integrating daily challenges, multimedia capture, and selective sharing within a trusted network, the prototype provides a practical alternative to conventional social platforms that prioritise screen time over well-being.

Through user-centred design, iterative development, and systematic testing, the project achieved its core objectives: delivering a functioning iOS application with challenge generation, media handling, authentication, and notifications. Usability testing confirmed that the interface was intuitive and aligned with user expectations, while backend testing established reliability at a prototype scale. These outcomes validate the central concept and establish a strong foundation for future development.

Nonetheless, limitations remain. The current system is constrained by local deployment, lacks advanced engagement features, and has not been validated at production scale. Addressing these challenges will require cloud migration, cross-platform support, and mechanisms for long-term motivation such as

personalisation and gamification. Future work may also explore the responsible use of AI to tailor challenges dynamically, though this must be balanced with rigorous privacy and ethical safeguards.

In conclusion, the Purpose App contributes both a conceptual and practical step toward rethinking how digital tools can support creativity, reflection, and social connection. While it is not yet production-ready, the project demonstrates that mobile technologies can be designed to promote healthier digital habits and more meaningful everyday experiences.

References

Books

Csikszentmihalyi, M. (2002). *Flow: The Classic Work on How to Achieve Happiness*. 2nd ed. London: Rider. ISBN 9780712657594.

Eysenck, M.W. (2012). *Fundamentals of Psychology*. 2nd ed. Hove: Psychology Press. ISBN 9781848721846.

Kaptelinin, V. and Nardi, B.A. (2006). *Acting with Technology: Activity Theory and Interaction Design*. Cambridge, MA: MIT Press. ISBN 9780262112987.

Nielsen, J. (1994). *Usability Engineering*. San Francisco: Morgan Kaufmann. ISBN 9780125184069.

Sellen, A. and Harper, R.H.R. (2002). *The Myth of the Paperless Office*. Cambridge, MA: MIT Press. ISBN 9780262692830.

Electronic Resources

Harvard Health Publishing. (2024). *The dangers of doomscrolling: How mindless scrolling impacts mental health*. [WWW page]. Available at: https://www.health.harvard.edu/blog/the-dangers-of-doomscrolling-2024 [Accessed 2 September 2025].

Kabat-Zinn, J. (2013). *Mindfulness in everyday life*. Greater Good Magazine. [WWW page]. Available at: https://greatergood.berkeley.edu/article/item/mindfulness_in_everyday_life [Accessed 2 September 2025].

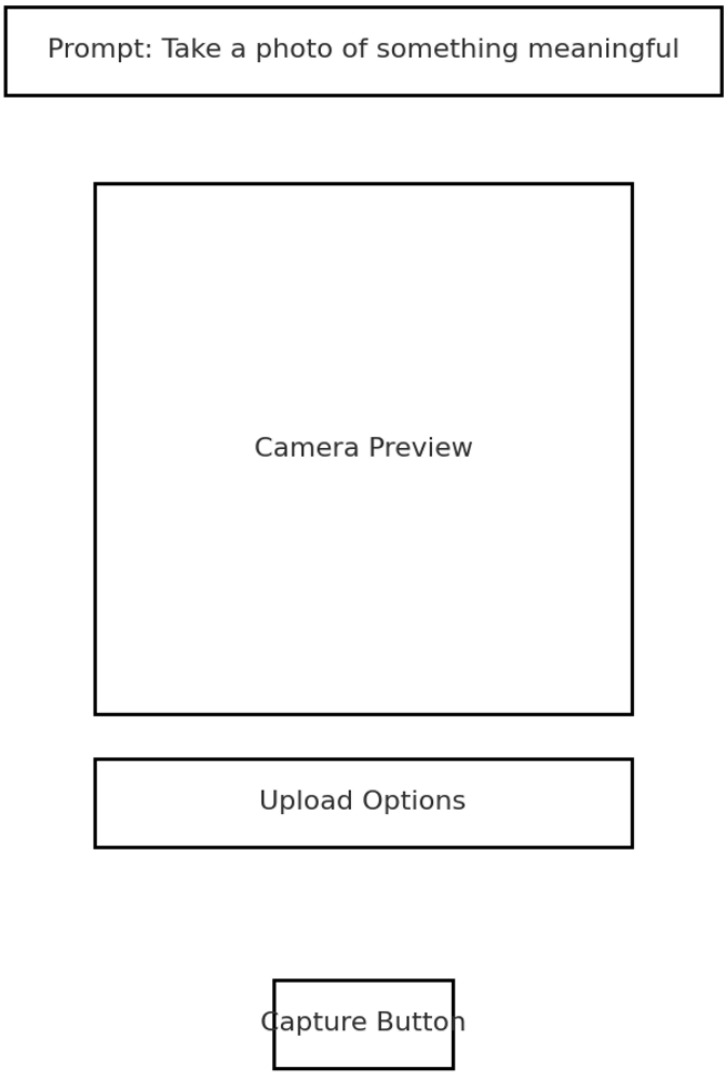
Mark, G., Gudith, D. and Klocke, U. (2008). *The cost of interrupted work: More speed and stress*. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. [WWW page]. Available at: https://dl.acm.org/doi/10.1145/1357054.1357072 [Accessed 2 September 2025].

Ryan, R.M. and Deci, E.L. (2000). *Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being*. *American Psychologist*, 55(1), pp.68–78. [WWW page]. Available at: https://doi.org/10.1037/0003-066X.55.1.68 [Accessed 2 September 2025].

World Health Organization (WHO). (2020). *Mental health and substance use: Stress and coping*. [WWW page]. Available at: https://www.who.int/news-room/fact-sheets/detail/mental-health-strengthening-our-response [Accessed 2 September 2025].

Appendices

Figure 1.1 Wireframe of Capture Screen

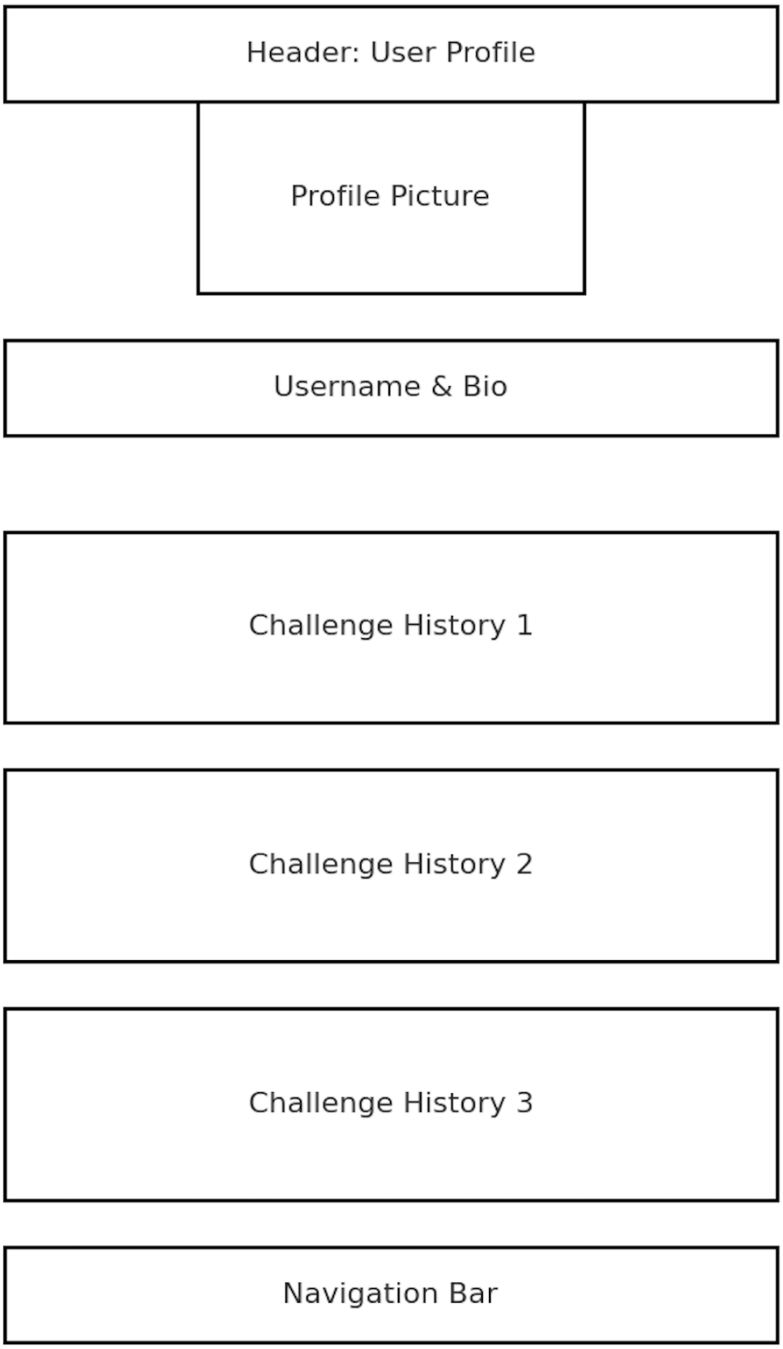


The Capture Screen provides users with their daily challenge prompt, displayed prominently at the top. A central camera preview allows the user to frame their photo or video, with upload options situated below to encourage flexibility. A clearly marked capture button enables rapid submission. This design prioritises simplicity and focus, ensuring that users can act on their challenge with minimal distraction.

Figure 1.2 Wireframe of Home Feed

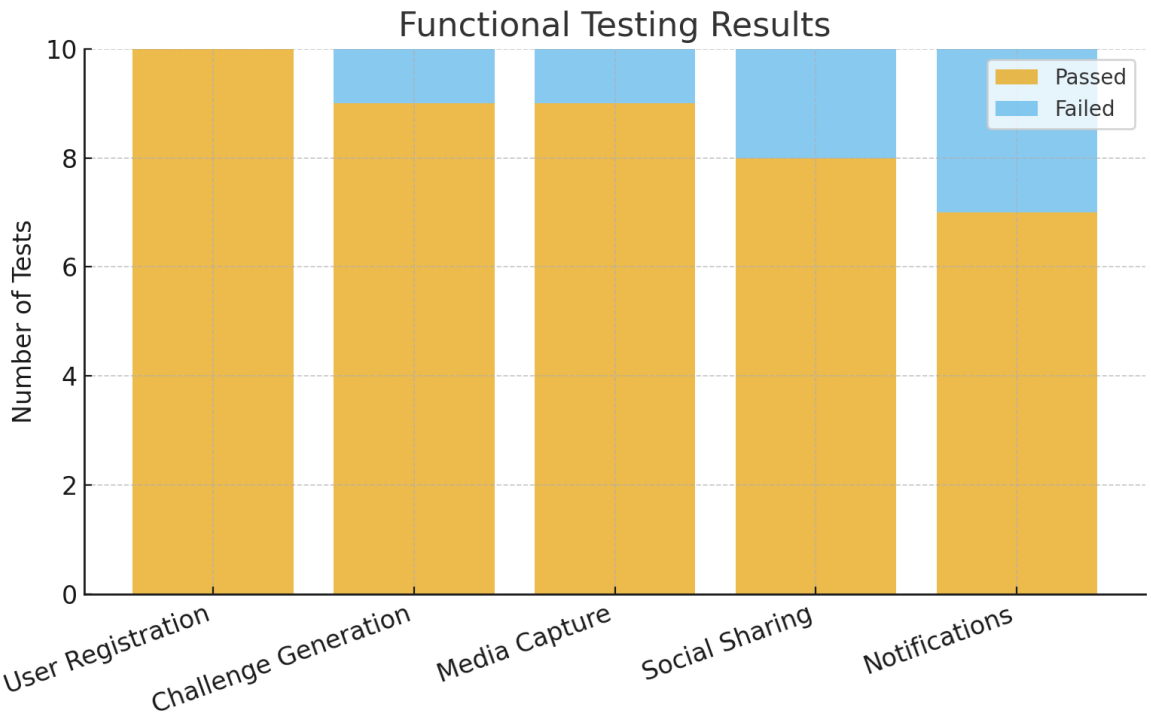
The Home Feed Screen showcases a curated selection of shared posts from the user's chosen community. Each post combines an image with a caption, fostering engagement and context. The layout is intentionally limited in scale to avoid the endless scrolling behaviour typical of mainstream platforms, instead encouraging users to view a small number of purposeful contributions. A navigation bar provides quick access to other areas of the app.

Figure 1.3 Wireframe of Users Profile



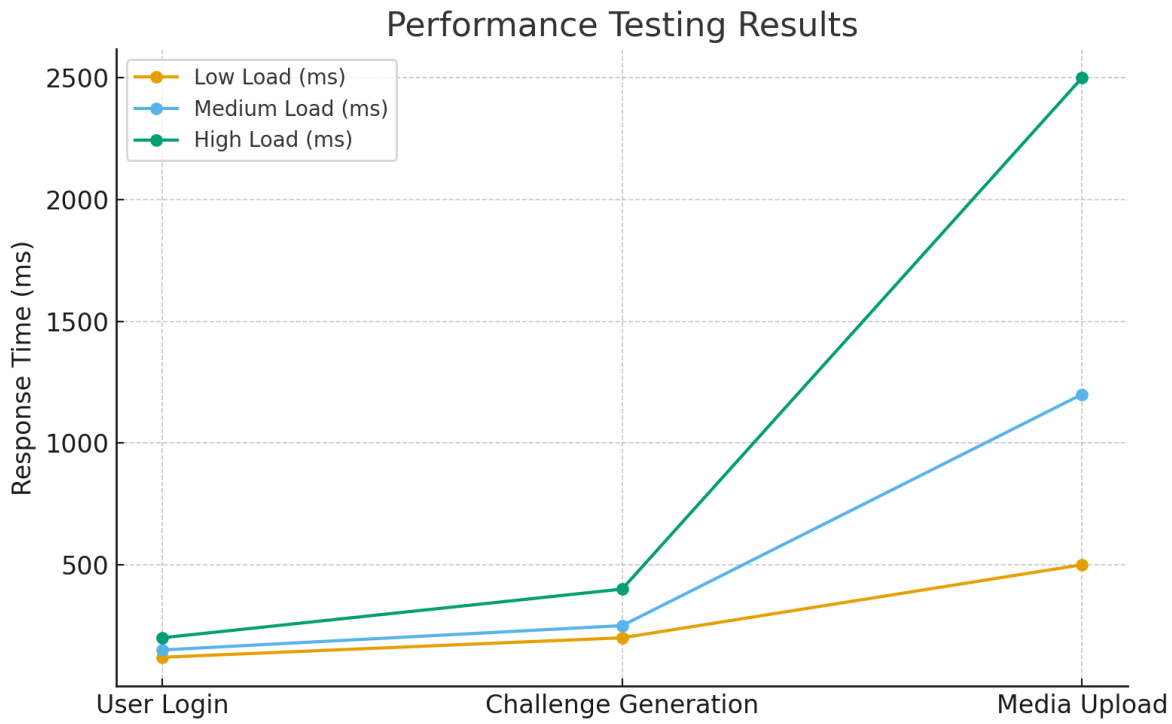
The Profile Screen displays key user details, including a profile image, username, and short biography. Below this, a chronological challenge history allows the user to review and reflect upon their past activities. The design reinforces continuity and progress, emphasising how small daily challenges accumulate into meaningful experiences. A bottom navigation bar ensures consistency with other screens, providing intuitive access to core functionality.

Figure 1.4 Functionality Testing



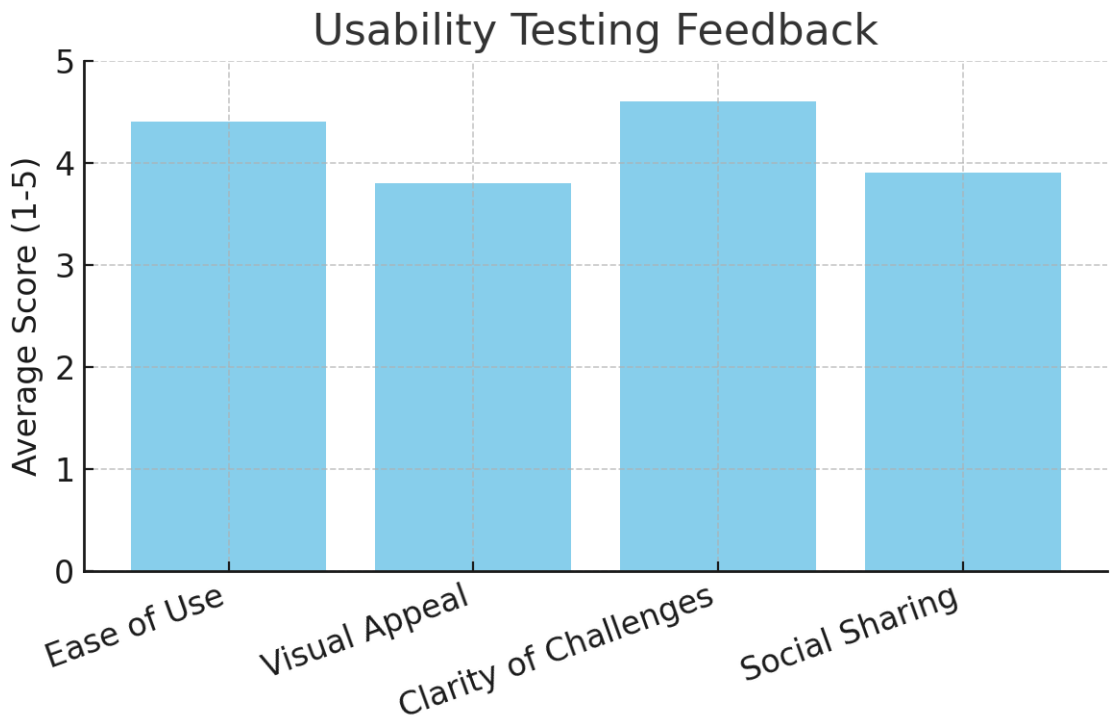
Functional testing results for the Purpose App. The bar chart summarises the outcomes of core features tested, showing pass and fail rates.

Figure 1.4 Performance Testing Results



Performance testing of core app operations under varying load. Results show average response times for user login, challenge generation, and media upload.

Figure 1.5 Usability Testing



Usability testing feedback from 10 participants. Average ratings (1–5) are presented for ease of use, visual appeal, clarity of challenges, and social sharing.