

DAILY – YOUR PRODUCTIVITY COMPANION

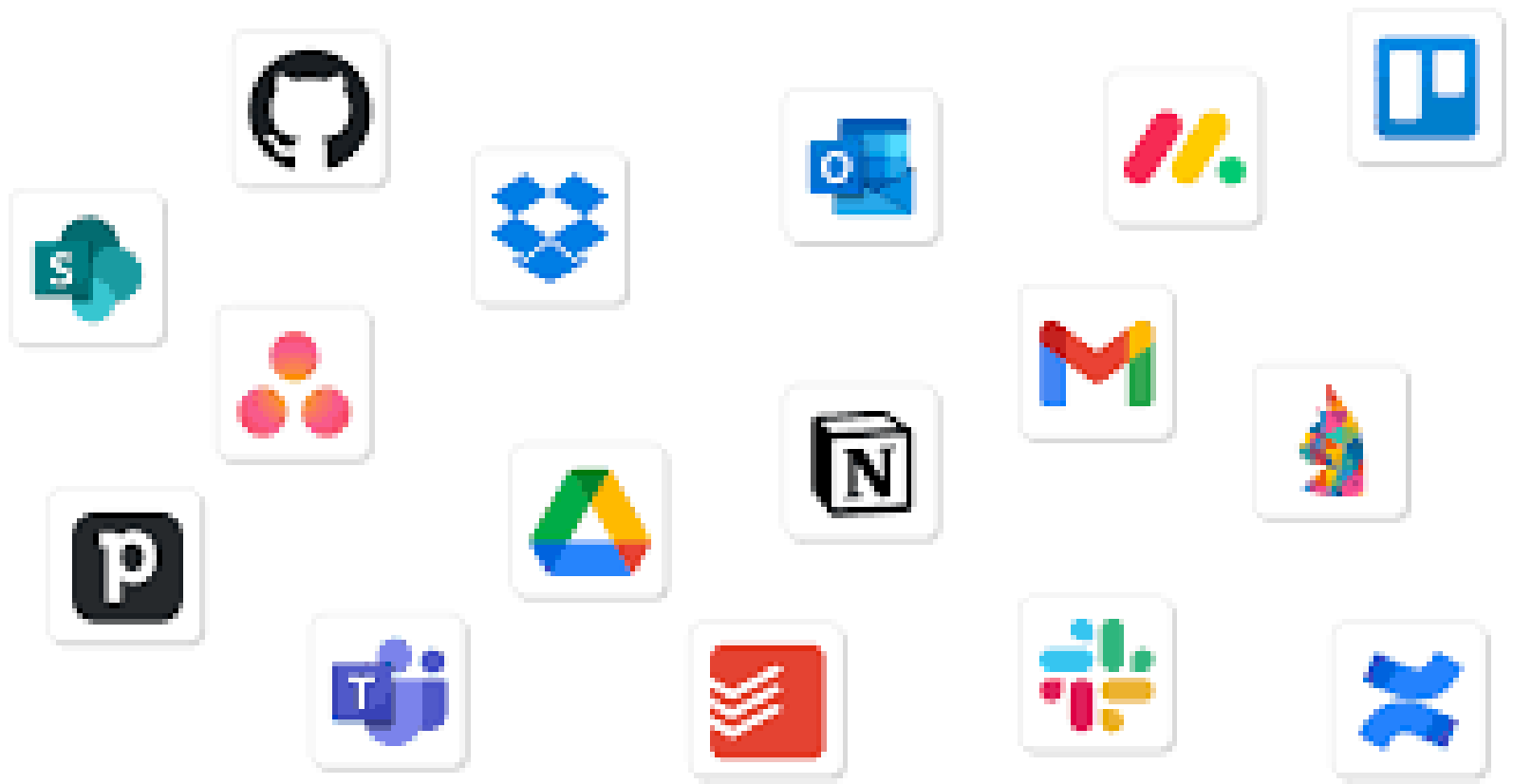
Group Members

16010123051 Anthony Jennis

16010123052 Anuj Madke

16010123053 Anurag Singh

PROBLEM STATEMENT



Modern users are overwhelmed by fragmented productivity tools that require multiple logins and interfaces.

There is difficulty in managing notes, to-do lists, and feedback through separate applications, leading to reduced efficiency.

Security and integration issues have prevented users from adopting a single, cohesive solution.

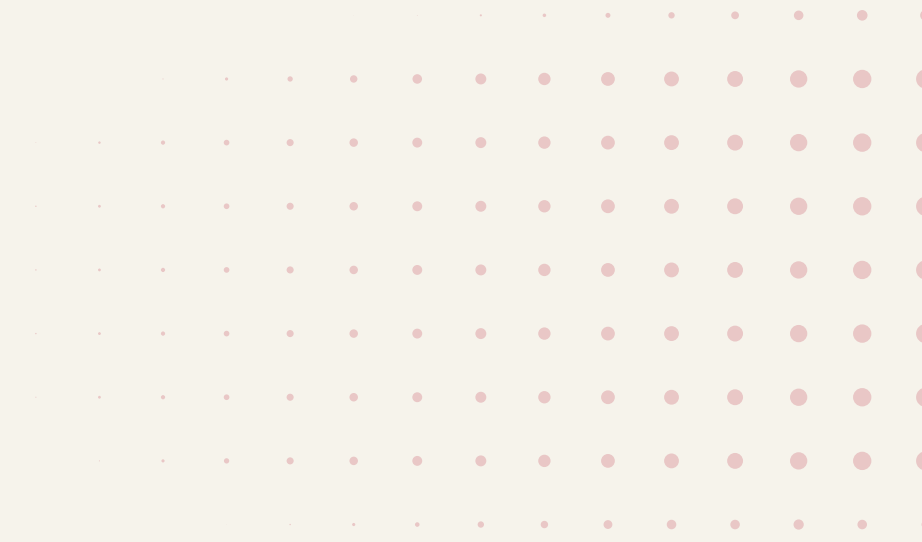


NEED STATEMENT

Unified Interface: Users require a single platform that integrates essential productivity features such as note management and to-do lists with secure authentication.

Ease-of-Use & Accessibility: A modern, responsive design should allow seamless task and note management across devices.

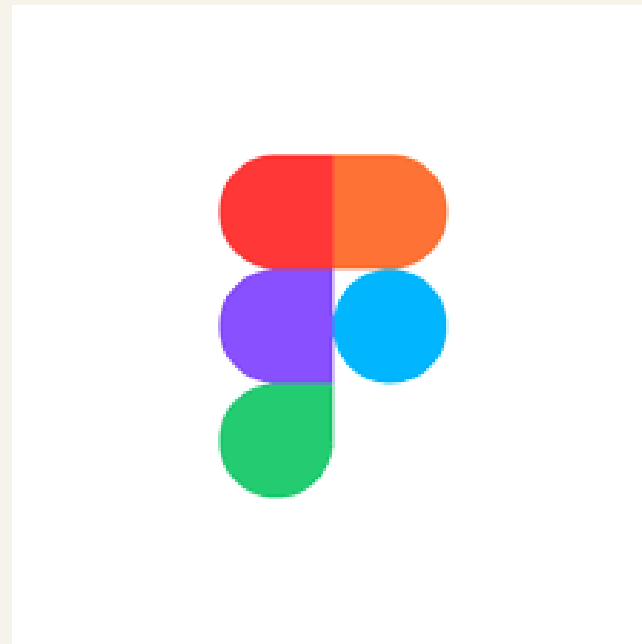
Security and Data Integrity: Robust user authentication and secure data handling practices are paramount for personal and professional use.



LITERATURE REVIEW

Evolution of Productivity Tools: Transition from desktop-based applications to integrated, cloud-enabled web applications

Web Technologies: The adoption of PHP, MySQL, and modern front-end frameworks has enabled faster and more robust application development.



User-Centric Design: Importance of responsive design and iterative feedback as demonstrated by Figma prototypes and usability studies.

Database Integrity: Emphasizes the necessity of secure database practices and prepared statements to prevent SQL injection



APPLICATION FLOW AND USER NAVIGATION

Login Flow:

Users begin at the Login/Signup screen.

Successful authentication leads to the Dashboard where personalized features become available.

Catalogue of Features:

Navigation includes To-Do (task management), Notes (note-taking with CRUD functionality), and Feedback (contact page).

Dynamic Menus:

Menus adapt based on session state to display relevant links (e.g., Sign Out when logged in, Login/Signup when not).



FUTURE SCOPE AND ENHANCEMENTS

Additional Functionalities:

Integration with calendar systems and file attachments for enriched task management.

Enhanced UI/UX:

Continued refinement of responsive design and mobile compatibility.

Progressive Web App (PWA) conversion for offline access and push notifications.

Security & Performance Enhancements:

Consistent use of prepared statements across all database interactions.

Implementation of centralized error logging and performance optimization.

Scalability and Integration:

Exploring cloud deployment and microservices architecture to better support a larger user base.

USER AUTHENTICATION (LOGIN.PHP)

```
if (password_verify($password, $row['password'])) {  
    $_SESSION['user_id'] = $row['id'];  
    $_SESSION['email'] = $email;  
    $_SESSION['created_at'] = time();  
    echo "<script>alert('Login successful'); window.location.href='index.php';</script>";  
}
```

This snippet demonstrates secure user authentication by using PHP's password_verify function.

Upon successful verification, session variables are set, ensuring that the user's state is managed across pages.

TO-DO MANAGEMENT (TODO.PHP)

```
if (isset($_POST['add_todo'])) {  
    $title = mysqli_real_escape_string($conn, $_POST['todo_title']);  
    $cat_id = intval($_POST['todo_category']);  
    mysqli_query($conn, "INSERT INTO todo (user_id, category_id, title)  
VALUES ($user_id, $cat_id, '$title')");  
    header("Location: todo.php"); exit;  
}
```

This code allows users to add new tasks under specific categories.

It sanitizes inputs using `mysqli_real_escape_string` and uses SQL queries to insert data into the tasks table.

NOTES CRUD OPERATIONS (NOTES.PHP)

```
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['save_note'])) {  
    $title = $_POST['title'];  
    $content = $_POST['content'];  
    $note_id = $_POST['note_id'] ?? null;  
    if (!empty($title) && !empty($content)) {  
        if ($note_id) {  
            $stmt = $conn->prepare("UPDATE notes SET title=?, text=? WHERE id=? AND user_id=?");  
            $stmt->bind_param("ssii", $title, $content, $note_id, $user_id);  
        } else {  
            $stmt = $conn->prepare("INSERT INTO notes (user_id, title, text) VALUES (?, ?, ?)");  
            $stmt->bind_param("iss", $user_id, $title, $content);  
        }  
        $stmt->execute();  
        $stmt->close();  
    }  
}
```

This snippet outlines the creation and update of notes using prepared statements for security.

It differentiates between a new note and an update operation, ensuring data integrity and protection against SQL injections.

CONDITIONAL NAVIGATION BASED ON SESSION (INDEX PAGE)

```
<?php if (isset($_SESSION['user_id'])): ?>
  <li><a href="todo.php">To-Do</a></li>
  <li><a href="notes.php">Notes</a></li>
  <li><a href="about.php">About</a></li>
  <li><a href="contact.php">Contact</a></li>
  <li><a href="logout.php" class="btn logout-btn">Sign Out</a></li>
<?php else: ?>
  <li><a href="about.php">About</a></li>
  <li><a href="contact.php">Contact</a></li>
  <li><a href="login.php" class="btn login-btn">Login</a></li>
  <li><a href="signup.php" class="btn signup-btn">Sign Up</a></li>
<?php endif; ?>
```

Session-Based Rendering:

This section is part of the navigation bar and displays different menu items depending on whether the user is logged in (determined by the existence of `$_SESSION['user_id']`).

Dynamic Content:

Logged-in users see links to additional features like To-Do and Notes, while guests are offered options to log in or sign up. This conditional rendering is key to a personalized user experience.

FEEDBACK SUBMISSION (CONTACT PAGE)

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = htmlspecialchars($_POST['name']);
    $email = htmlspecialchars($_POST['email']);
    $message = htmlspecialchars($_POST['message']);

    if (empty($name) || empty($email) || empty($message)) {
        echo "<script>alert('All fields are required.');
```

Sanitization:

This snippet starts by sanitizing the received POST data using `htmlspecialchars`—protecting against cross-site scripting (XSS) by escaping any special characters.

Validation:

It checks that none of the required fields (name, email, message) are empty before proceeding.

FEEDBACK SUBMISSION (CONTACT PAGE)

Database Insertion:

If the input is valid, the script constructs an SQL query to insert the feedback into the contact table.

Note that while this approach uses basic input sanitization via escaping, in a future revision using prepared statements consistently would further enhance security.

User Feedback:

The code uses JavaScript `alert()` calls for immediate user feedback regarding success or failure, and then redirects back to the same page.

CONCLUSION

"Daily" successfully integrates core productivity functionalities into one cohesive web application.

The application addresses critical user needs by streamlining note taking, task management, and feedback into a unified platform.

Secure authentication, modular page flows, and a user-friendly interface ensure that the application is both robust and accessible.



THANK YOU