

MG-Monkeys Team Project

Team Name & Members

MG-Monkeys

Team Members:

- Team Lead - Nate Brewer
- Frontend Developer - Jack Dixon
- Backend Developer - Danny Poff
- Database Engineer - Philip Bierley

Problem Domain

Web-Based To-Do List Application

Problem Statement:

Users struggle to effectively manage and organize their daily tasks in an entertaining way. There is a need for a simple, intuitive, and accessible solution that allows users to:

- Create and manage tasks efficiently
- Organize tasks by categories or priority
- Access their tasks from anywhere
- Collaborate with team members

Features & Requirements

Key Features (8 Total)

1. To-Do Tags - Categorize To-Do's

- DB tag collection
- Searching/filter for tag's
- link between tag and post in the DB
- Tag-validation

2. Front-End User Customization - Make your list your own

- Change fonts/font colors
- Add primary, secondary, trutiary colors
- Look for text editor packages to add

Features & Requirements (Cont.)

3. **AI-Generated Tags** - Utilize External GenAI to create tags automatically for posts

- External genAI api to create tags
- Tag validation (Non-expensive)

4. **Browser Notifications** - Reminder!

- Determine user browser of choice
- Hit the specific browsers API to send notifs
- User-set notification time

Features & Requirements (Cont.)

5. Chatbot with Mascot - Monkey-See, Monkey-Do

- GenAI API for the chatbot.
- Interact with to-dos (Celebrations/negative reactions to completion status)
- Create per-page popup that the user can interact with
- Interface with the to-do's to create reactions.

6. Recurring To-Dos - Schedule repeating tasks

- Create Weekly/Monthly reminders

7. Email-SMS Notifications - Know at a glance

- utilize external package Email service to send user reminders
- (if using sms - find an package for that)

Features & Requirements (Cont.)

1. Group to-do list - Team projects

- Non-authed p2p invitation
- Database collection of the group
- Accept/Decline invitation page/popup
- Group Page (re-skin of individual page)
- Group to-do assignment: 2 approaches
 - Who ever created the page will assign to-dos and create them
 - Non-hierarchical approach, users will create to-dos and have a "liking" system that will similar to assignment

Features & Requirements (Fin)

Total Features 8

Total Requirements 26

Development Roadmap

Sprint 1 Deliverables

1. Adding **tags** to the to-do list items
2. Front-End **user customization**

Sprint 2 Deliverables

3. Using **AI** to automatically **generate tags**
4. Adding **notifications** to the browser
5. (Start) Create a **mascot** that also acts like a **chatbot** (Monkey-see Monkey-do)

Sprint 3 Deliverables

- 6. (Start) **Group to-do list** for team projects
- 7. **Recurring to-dos**

Sprint 4 Deliverables

- 8. (Finish) Create a **mascot** that also acts like a **chatbot** (Monkey-see Monkey-do)
- 9. (Finish) **Group to-do list** for team projects
- 10. **Email notifications** (SMS - Maybe)

Team-Rules

1. **Communication** is key
2. **Leader will update team page weekly**, while team members will update their **individual progress page weekly**.
3. **Weekly meetings - in-person Tuesdays at 1pm. - Discord will be the backup and primary online communication pipeline.**
4. **Team members will use their own branch to complete their work. When feature is completed, create a pull-request, and the leader will test and review it.**

Questions?

Thank you for your attention!

Project Repository: [GitHub Link](#)