



Final Project

TASK IT

420WE6AB Web Development I

Iana Setrakova (2457647)

Table of contents

Planning and Preparation	2
Planning, steps performed	2
Wireframe	2
Data Model	3
E/R diagram	4
Target Audience	6
Research performed	7
Website Details	11
Git Hub Repository URL	11
Overall Goal and Purpose	11
Description and Features.....	11
Workflow Overview:.....	11
Communications	13
Post-Mortem	17
Project Feedback	17

Planning and Preparation

Planning, steps performed

Our plan includes to choose the optimal design comprising essential features like user registration and task management. The webpages to be created are:

- Home page
- User log in
- Users sign up
- Dashboard to manage user profile and tasks
- Page showing a single task to be created or updated
- Contact us page

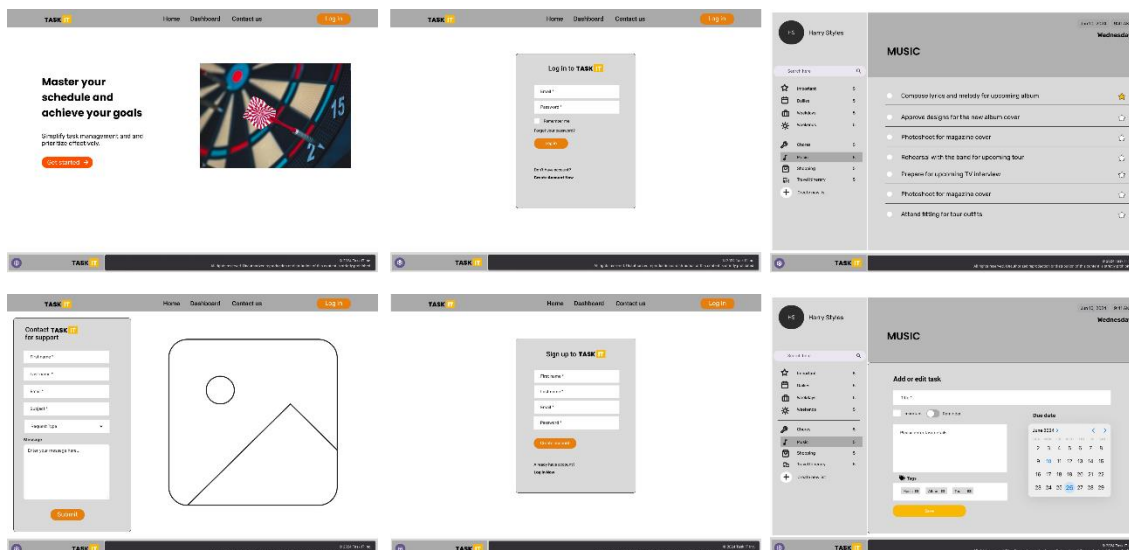
We need to enable PHP session when user logs in. And when session is created, user can

- Manage own profile
- Delete their account
- View the listing of to-do items
- Adding new to-do item
- Updating items
- Removing items
- Completing item

Wireframe

Wireframe was done in Figma:

<https://www.figma.com/design/8hCZa7zYgBmKGfqnXkQFc6/design-cv-and-tasks?node-id=0-1&t=EmtJsFLZp9umCcrz-1>



Data Model

- **Users:** Can create multiple tasks, each associated with only one user.
- **Tasks:** Belong to one list.
- **Lists:** Can have multiple tasks.
- **Reminders:** Each task can have one reminder.

EXTRA possible features for future development:

- **Tags:** Associated with one or multiple tasks.
- **Task Collaborators:** Enables multiple users to collaborate on a single task.

Database structure in Third normal form (3NF):

users = (id (PK), username, password, avatar_image, email, registration_date)

tasks = (id (PK), title, description, task_order, *user_id (FK)*, *list_id (FK)*, creation_date, due_date, completed, importance)

lists = (id (PK), title, user_id, list_order)

Extra tables for future development:

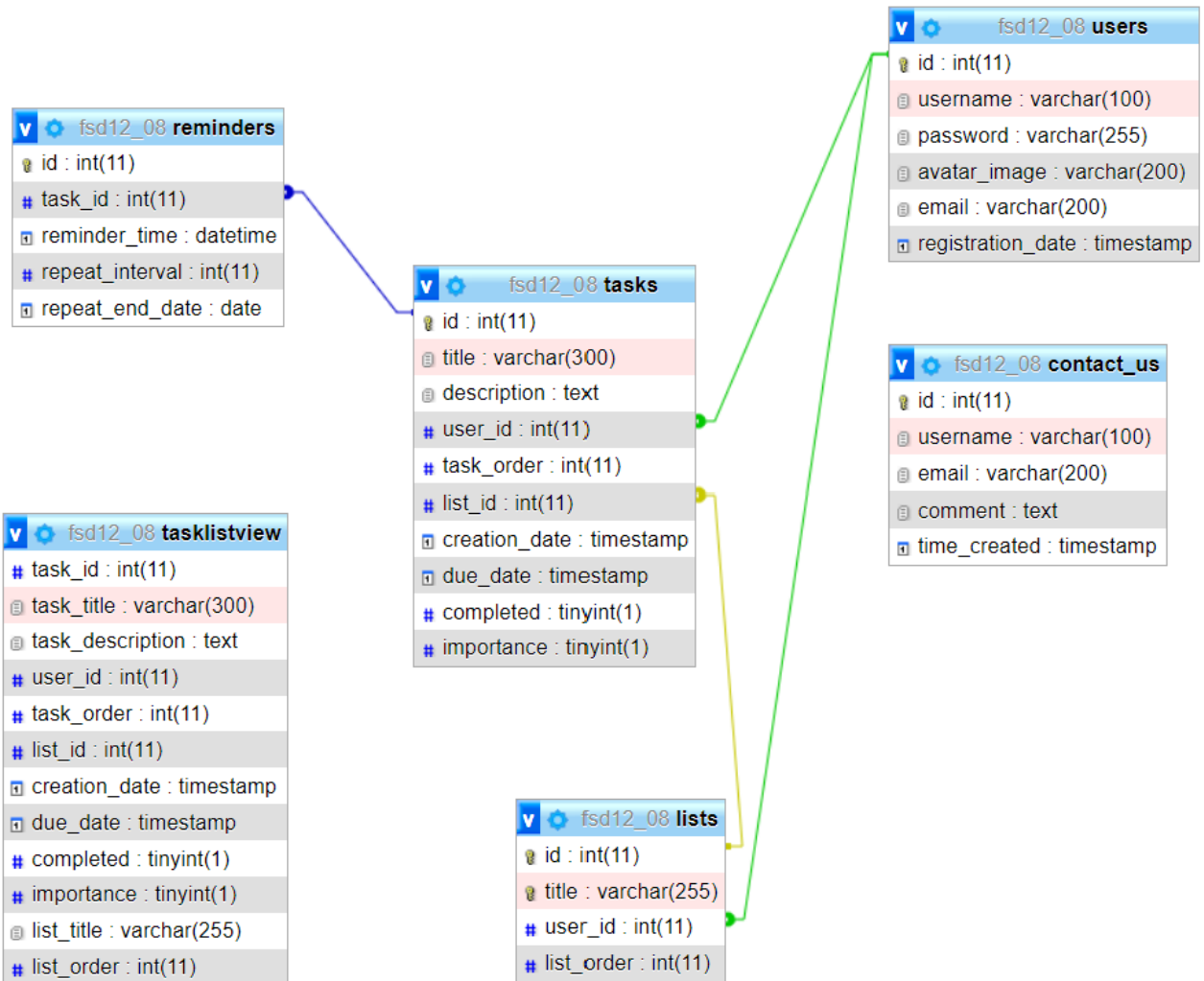
reminders = (id (PK), *task_id (FK)*, reminder_time, repeat_interval, repeat_end_date)

tags = (id (PK), name, tag_color)

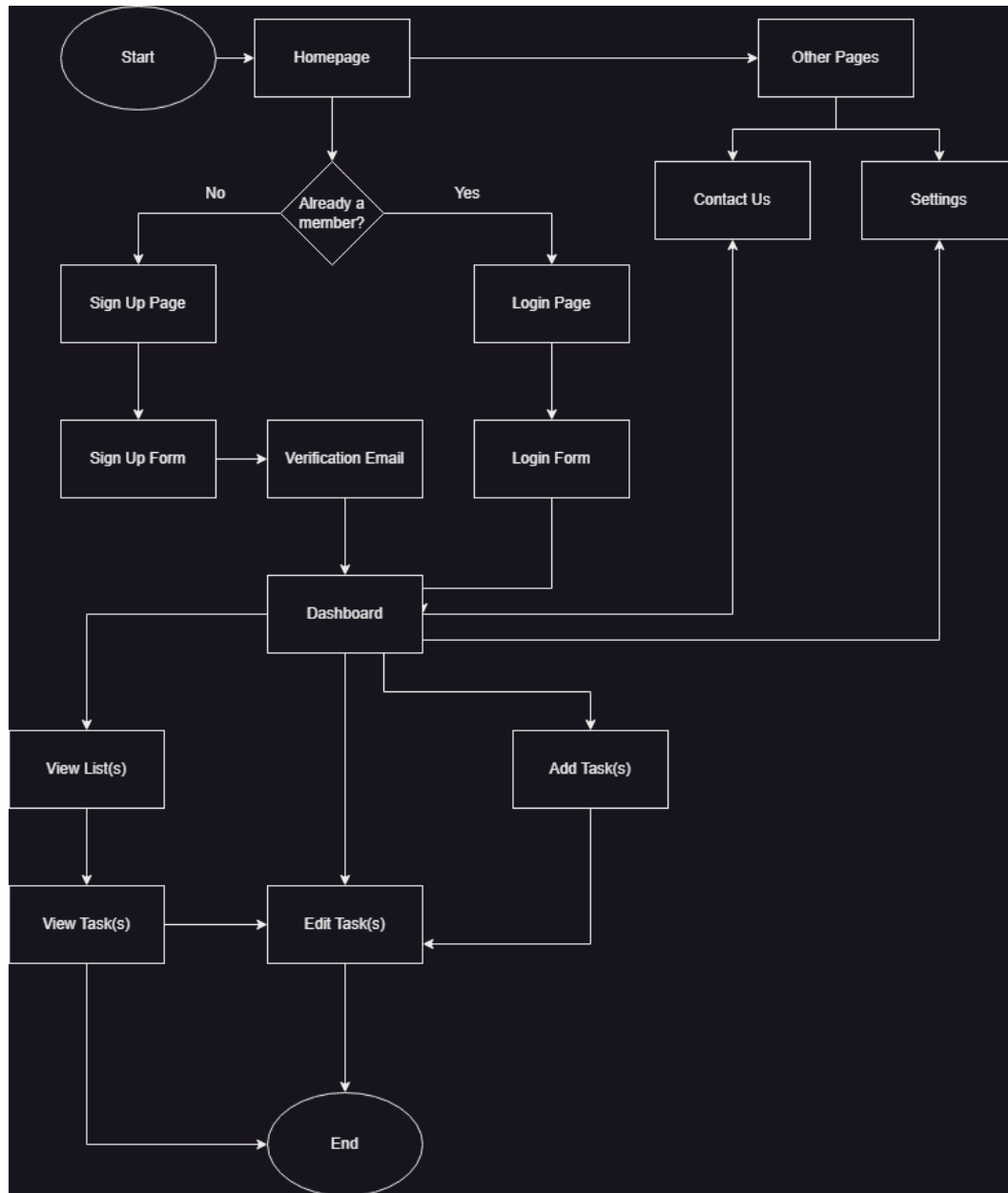
task_tags = (*task_id (FK)*, *tag_id (FK)*)

task_collaborators = (id (PK), *task_id (FK)*, *user_id (FK)*)

E/R diagram



Flowchart (preliminary)

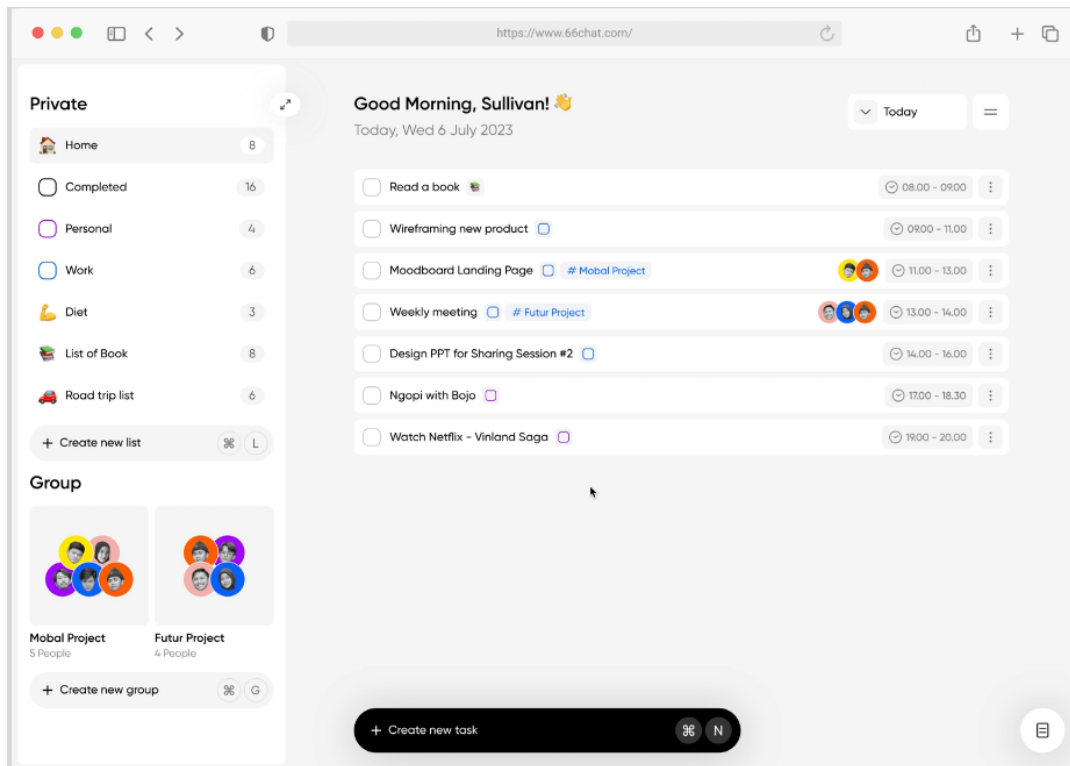


Target Audience

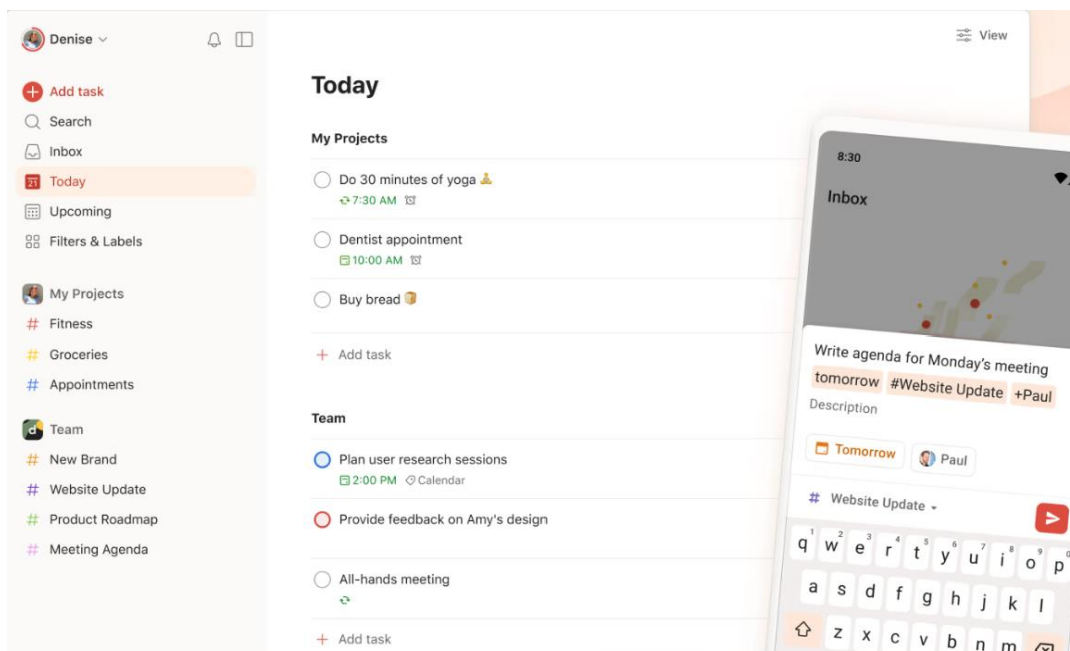
1. Students:
To organize homework and study schedules.
2. Professionals:
To manage work tasks and project deadlines.
3. Home Organizers:
To keep track of household and family activities.

Research performed

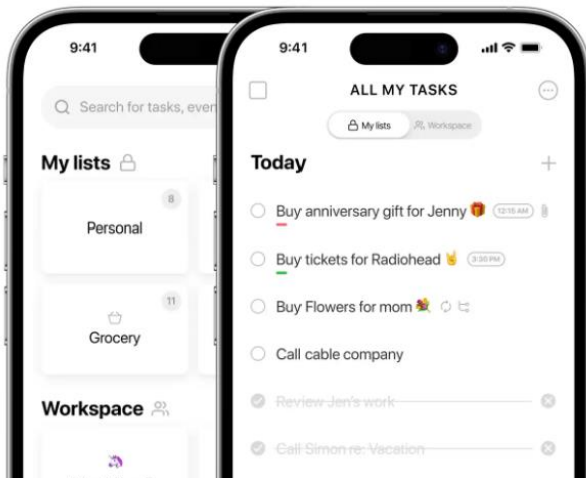
<https://dribbble.com/shots/22373497-Create-a-task-Interaction>



<https://todoist.com/>

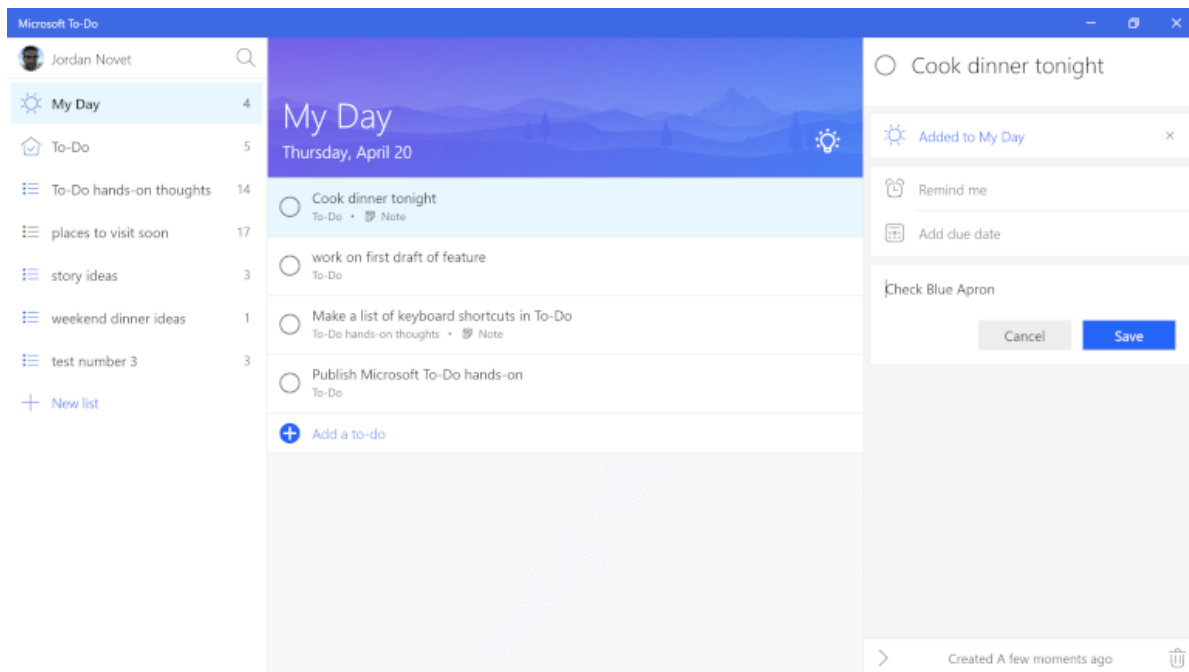


<https://www.any.do/>

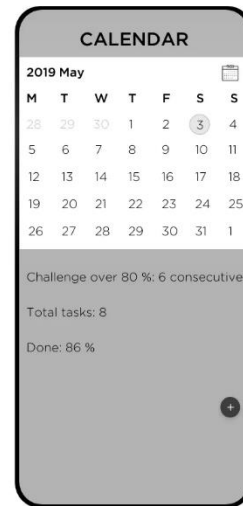
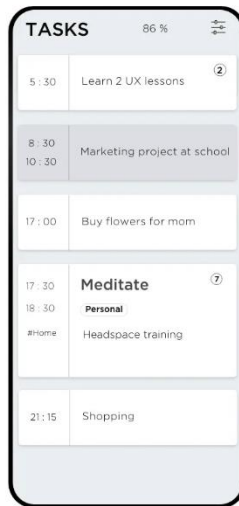
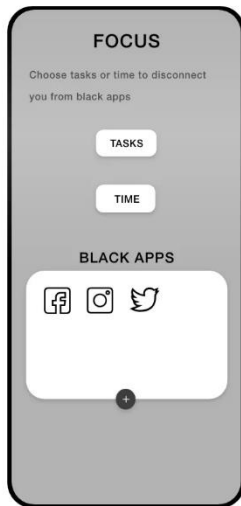
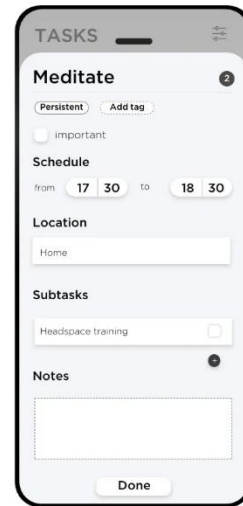
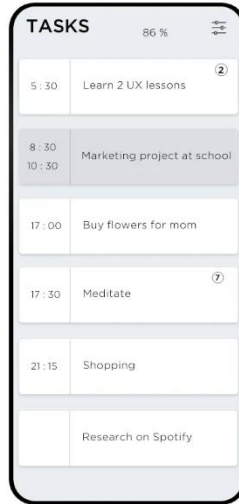
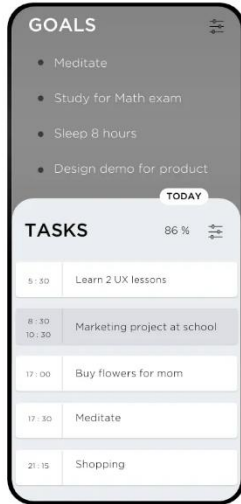


Microsoft To Do

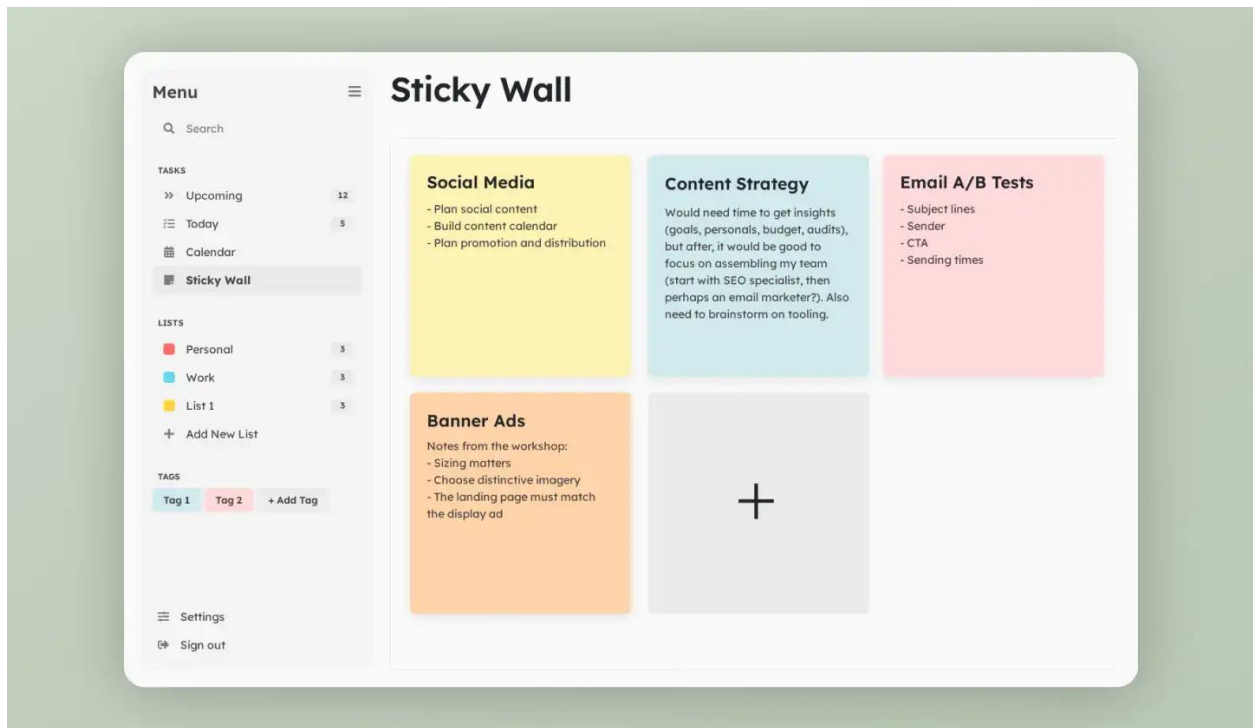
<https://to-do.office.com/tasks/>



<https://uxdesign.cc/got-stuck-with-to-do-list-i-create-a-new-app-819879fedce0>



<https://uizard.io/templates/website-templates/to-do-website/>



Website Details

Git Hub Repository URL

<https://github.com/iasssy/web-development-8>

Overall Goal and Purpose

The website aims to provide users with a comprehensive To-Do list application that helps them manage tasks efficiently. It allows users to create, edit, and delete tasks and lists, and offers features like task importance, quick statistics, and a user-friendly interface.

Description and Features

The website includes key pages such as the homepage, login, signup, dashboard, and contact us. Users can create accounts, log in, and manage their tasks from the dashboard. The application supports AJAX for seamless task management and dynamic content updates. Future features will include user profile modification, list order customization, and due date handling.

Workflow Overview:

- **Day 1:** Researched designs, documented the data model and E/R diagram, described the target audience.
- **Day 2:** Created wireframes for the homepage and dashboard in Figma, designed the logo, and built login and signup pages.
- **Day 3:** Completed the homepage, login, and signup pages, exported wireframes, and started documenting the initial workflow.
- **Day 4-14:** Developed the website structure with the Fat-Free Framework, created routes, and implemented dynamic pages and AJAX functionality for task management. Worked with form validations, user session handling, and error management.

Requirements

I met the requested requirements of my To-Do list application project by following a structured approach. I started by researching design options, creating wireframes, and documenting the data model and E/R diagram, ensuring it was normalized to 3NF. All tables, attributes, and relationships were properly analyzed and created. Sample data was included in the final database to make the site functional upon initial load.

During development, I used the Fat-Free Framework for organization and created routes for easy navigation. Essential pages like the homepage, login, and signup were developed with proper form validations. Features for creating, editing, and deleting tasks and lists were implemented using AJAX for dynamic updates.

Communications

Day 1: 2024-07-09

- We got acquainted with team member's background.
- The tasks divided after the first meeting:
- Iana begins researching various To-Do List websites to gather insights for our project's database design, makes 3NF of database.
- Craig starts research on specific functionalities and started drafting the storyboard and flowchart to visualize the website's structure and user interactions.
- create GitHub repository
- created shared word document on teams
- created collaborative Figma space

Day 2: 2024-07-10

- Worked together on creating a mock-ups for the application's dashboard
- added name + logo
- hashed out some features we'd like, customizable icons, daily, weekly to-do lists, recurring, etc.
- divided up some tasks as well, Craig: contact us page; Iana: sign up/login page

The team is now composed of one person, so there will no longer be group progress updates, only individual progress logs.

Individual progress log

Day 1 (2024-07-09)

- Made research on possible designs
- Started to fill out documentation concerning Data Model and E/R diagram
- Described Target audience
- Filled out documentation about research

Day 2 (2024-07-10)

- Started to create a wireframe for Homepage (Landing page)
- Worked together in Figma creating wireframe design of Dashboard page
- Created logo
- Created login and sign up page

Day 3 (2024-07-11)

- Finished log in, sign up, homepage pages
- Worked with Figma wireframe together
- Exported all the wireframes
- Started "Initial Workflow.txt"

Day 4 (2024-07-12)

- Created folder for website, installed Fat-free framework, created necessary folders
- Created branches
- Decide to work on building login and sign up pages with Bootstrap

Day 5 (2024-07-14)

- Create .htaccess file and routes for homepage, login, Dashboard
 - Created initial design of home page and log in and set up the title and meta description to be retrieved as variables
- from class Pages

- added default title, made default title to be dynamically appended with current title

Day 6 (2024-07-15)

- Continue working on sign-up page
- Start working on dashboard
- Understanding further Fat-Free framework

Day 7 (2024-07-17)

- added contact us page, created routes for sign up and contact us page
- started the creation of dashboard page
- created the homepage illustration image

Day 8 (2024-7-18)

- almost finished dashboard page, added JavaScript links to footer, but couldn't change header background,
- corrected links on the button on hero banner on homepage, added input names to sign up and log in
- changed f3 to protected and removed from functions, added signupSave route
- added config.ini, access.ini files, added ListController, classes Model, Lists, Tasks, User,
- created lists.html and tasks.html (the last one is still not functional)

Day 9 (2024-7-19)

- added ContactUsController, corrected dashboard rendering
- created taskController, TaskListViewController for database View "tasklistview"

Day 10 (2024-7-20)

- added contact us form proper validation with additional functions and repopulation of data
- sign up page form also works to save to database (still needs validations)

Day 11 (2024-7-22)

- fixed function fetchByName by including tableName
- corrected the way the errors are shown
- worked on sign up page and function checkRememberMe (for setting the cookie with automatic user log in)
- fixed access ini file, and added it to GitHub (with 'vendor' and 'temp' folders)
- added session start and checking it when rendering dashboard, finished log in and sign up pages validations (also did log out)
- started to work on dashboard, fixed footer on small pages to go to bottom

Day 12 (2024-7-23)

- created functions addList and addListSave, showListByIdWithTasks
- in class Model created fetchTableByColumnValue
- created ajaxes for rendering form and submitting of new List, and for rendering left side list upon adding a new list
- added text-overflow: ellipsis for the list to be shown partially on smaller devices
- also showing user name and avatar if it is in database

Day 13 (2024-7-24)

- Created functions addList, editList, deleteList, addTask.
- Handled ajax requests on re-rendered items

Day 14 (2024-7-25)

- Handled changing status of completed task, their importance, with counting their statistics.
- Handled editing of task, deleting. Fixed the redirecting of all form submits.

Post-Mortem

There were errors that I was unable to fix:

- For some reason, when the user has no lists, a new list is not created, but it worked at the beginning, I cannot fix it right now
- For the logged user, if the list has no tasks, where the quick statistics panel is not present, and the jquery of updating it (updateQuickStats) is still performed. I tried to add it (having "0"), but couldn't for some reason. That's why the Console shows jquery errors.
- gitignore seems not working, still sending "vendor" and "tmp" to GitHub
- For some reason, if I use Fat-Free syntax in "tasks/edit-task.html", this form stops working, had to use simple PHP

Features that will be implemented in future:

- checkbox "Remember me" functionality in both log in and sign up
- user profile modification (username and avatar)
- list order implementation
- due date wasn't worked with

Project Feedback

The idea of having a project developing a To-Do list application is practical and. It is a good learning process to work with frameworks, by fixing the issues and understanding the logic. The main problem I had with this project was inability to find the reason why pages or some features stop working and there was no way to find what it was. Sometimes it was just typo in class name, or route variable naming different in two parts. At least I understood, I can use Dev tools with Network with preserved log and error_log file where I can print out something for debugging.