

Command T



Software Requirements Specification

Authors:

Yejin Shin - yejin.shin@stonybrook.edu

Zhanarbek Osmonaliev - zhanarbek.osmonaliev@stonybrook.edu

Hasung Jun - hasung.jun@stonybrook.edu

Fabio Calero - fabio.calero@stonybrook.edu

Table of Contents

1. Introduction	3
1.1 Product Description	3
1.2 Scope	3
1.3 Users	4
1.4 User Feedback	4
1.5 Existing Alternatives	5
1.6 Definitions	6
1.7 References	6
2. Requirements	7
2.1 Functional Requirements	7
2.2 Use Cases	7
2.2.1 Use Case: Add a bookmark	7
2.2.2 Use Case: Customize layout of the objects	8
2.2.3 Use Case: Add a note using Notes widget	8
2.2.4 Use Case: Add a task using Todo List widget	9
2.2.5 Use Case: Move a bookmark into another folder.	9
2.2.6 Use Case: Delete Objects	10
2.2.7 Use Case: Add bookmark using the extension	10
2.2.8 Use Case: Change the background image	11
2.3 User Interfaces	12
1. Landing page	12
2. App homepage	14
3. Widgets	18
4. Settings	20
5. Guidelines	26
6. Use Extension	27
2.4 Non-functional Requirements	27
3. Contributions	28

1. Introduction

1.1 Product Description

Using modern-day web browsers to save and organize access to web links has become a rather messy and cumbersome task. Safari, Chrome, and Firefox have very poor options for managing and organizing bookmarks. When users add bookmarks, they usually appear unorganized and spread without any meaningful layout. We propose a web application that will serve as a replacement for the Safari, Chrome, and Firefox homepages and allow users to customize their startup page, organize their bookmarks, make todo lists, and add customizable widgets, and take notes all in one place.

Our web app, CommandT, aims to create a webpage that serves as a startup page of any internet browser. Our targets are students and professionals from all fields who want to customize their startup page. CommandT will allow users to manage their bookmarks by creating folders, divide sections, categorize, and more. It will also contain widgets for notes, todo lists, and weather. Users will need to sign up for CommandT to keep their settings next time whenever they access the application again.

Main Features:

1. Managing bookmarks with folders.
2. Allow users to customize order of bookmarks and widgets (todo list, weather, and notes).
3. Allow users to customize the application's background image.
4. Use built-in Notes and Todo-list widgets.
5. Make extensions for Chrome and Firefox.

Stretch Goals:

1. Sidebar for quick actions such as chat for teams.
2. Calendar widgets.
3. Users will be able to share their todo lists with other users.

1.2 Scope

This project intends to provide an effective and quick way for people to organize their bookmarks on modern browsers. The focus is on allowing the user to customize their startup pages by using advanced options that are not available by default on most browsers. It is not our project's purpose to substitute the in-built startup pages on modern browsers but rather to allow our users to control the content they see by providing an alternative. Furthermore, it is not our project's goal to track our user's daily activity nor to automatically add bookmarks for our users. Instead, each user will opt to save their bookmarks by manually entering them through our web application.

The product will be implemented as a responsive web application that will be usable on most modern desktop computer web browsers, specifically Safari, Chrome, and Mozilla Firefox. Our application will not support mobile browsers.

1.3 Users

Our project's primary users are university students who want to increase their productivity and efficiently organize their online class links and adult professionals from all fields whose work might involve visiting several websites regularly. We expect both of these user groups to be somewhat knowledgeable on the basic use of a browser and be somewhat familiar with technology in general.

Although our project primarily focuses on the above-mentioned audience, it will be available to a general audience as well.

1.4 User Feedback

Our goal is to target individuals and teams initially. Therefore, we have presented our mockup for Command T to ZeroClassLab, a SUNY Korea startup company. For feedback, we consider the ZeroClassLab a team instead of a company since they only have five employees at the time. Our mockup presentation informed us that the website does not have enough functionality to dramatically enhance their team's productivity due to limited widgets. The feedback we received from ZeroClassLab made us reevaluate our target users. As a result, we choose to initially target individuals rather than individuals and teams. The functionalities we will offer are not enough for teams to use because teams require widgets like calendar widgets that allow a team to share meeting day and time, holiday, and upcoming events as a scratch goal.

We also received design feedback. They asked how we will handle a situation when the text is invisible from setting a background image that has similar color to text color. We decided to handle the situation by changing the text color to a color that contrasts with the color of the background image.

After the interdisciplinary project pitch, a few professors and students were willing to use our Figma prototype and give feedback as alpha-testers. We will gather about 4~6 more testers to receive feedback. We plan to have the testing via Zoom. When the users enter the zoom session, they will get a link to our Figma prototype and interact with it. They will be asked to give feedback right after they have fully experienced our prototype.

The last feedback will be after the full implementation of Command T. We will first ask them, "Does our final product match the requirements and design?". Then we will gather about ten beta-testers from our possible users stated in 1.3 Users. We will ask them several questions, including "Do you think Command T is ready to be released?", "Any more complement is needed?". Based on their responses, Command T will go through revision or be released.

1.5 Existing Alternatives

Existing alternatives to our product include the built-in bookmark managers from Chrome, Firefox, and Safari, as well as several bookmark manager applications found on the web. We feel that our product brings something new to the table. Therefore, we detail our main competitors and the shortcomings in their product that we think our project addresses.

1. Atavi: a free bookmark manager that allows users to create a customized homepage with bookmarks presented in an elegant and organized fashion. Atavi enables users to organize bookmarks into folders and to sync their bookmarks with any device or browser. It allows the users to customize almost everything from the background picture to the number of bookmarks on the page. Most of the inspiration for our product comes from Atavi. Effectively, we have tried to make a better improved Atavi with additional features to increase ease of use and productivity.
2. Raindrop.io: a visually enticing bookmark manager that allows its users to organize bookmarks into collections which can be displayed in headline format, card format, and mood board format. This application automatically tracks how frequently the user accesses a specific bookmark and groups bookmarks by frequency of use, as well as grouping the broken links in one place for the convenience of its users. Raindrop.io also allows users to save more than just bookmarks on their cloud accounts. It allows users to save pictures or to upload files into their collections. Raindrop.io compares to CommandT in so much as they are both bookmark managers. However, the philosophy behind our product and the design, and the feature set are completely different.
3. Bookmark Ninja: a cross-platform bookmark manager with extensive features, including a tab system for organizing bookmarks and the ability to tag bookmarks and then sort through bookmarks by tag. Booking Ninja is a very well-rounded app, but it differs from our product in that our product offers simplicity, where Bookmark Ninja is complicated. The many tools in Bookmark Ninja make it hard to learn to use for the uninitiated while our product strives to be intuitive and user friendly for all users, an idea that was inspired by Apple's philosophy of simplistic design.
4. Save to Pocket: a very simplistic yet popular web app for managing bookmarks. It allows its users to add bookmarks into lists that can be nested and annotate each bookmark. Its main shortcoming is that it is not customizable and the list can become cluttered very easily, making it hard to manage a large number of bookmarks effectively.
5. Google Bookmarks: a cloud-based bookmark managing application that comes as an extension for google chrome. It allows its users to save bookmarks to their google accounts, add labels and notes to each bookmark. Google Bookmarks doesn't let users put bookmarks into folders, and the interface is rather unattractive and not designed to be a homepage in the user's browsers. Note: Google Bookmarks is not to be confused with the chrome built-in bookmark manager as it is a separate service.
6. Pinboard: the oldest bookmark manager in this list and most likely the one with the largest user base. It has a rather old look that takes no consideration for the user experience.

On top of the applications listed above, there are various note-taking apps that offer bookmark organizing features such as Evernote, Notion, and One Note. These apps offer a fast clipper tool and allow its users to manage their bookmarks into folders. However, the process is relatively slow due to the number of steps it takes to save bookmarks, it is not customizable, and the applications are not desktop friendly since these applications were originally designed to be mobile and tablet apps.

1.6 Definitions

- **Object:** refers to widgets, folders, and bookmarks.
- **Start up page:** the page displayed when users open up a new tab.
- **Layout:** the distribution of widgets on the screen.
- **Widgets:** a component of the interface that provides a service to the user (e.g. the weather widget: a component that provides weather information)
- **Web App:** short for “web application” meaning an application that is hosted on a web server.
- **Browser:** an application that allows the user to surf the web and display websites (e.g. Google Chrome, Mozilla Firefox, and Apple’s Safari).
- **Bookmark:** an entry or icon that effectively serves as a shortcut to a website the user has previously visited.
- **In-built:** available by default.
- **Todo list:** a list of tasks set to be performed at a later time by the user.
- **Todos:** several tasks belonging to a todo list.
- **Extension:** a small module that can be added to a browser to customize the use of the browser.

1.7 References

- iOS 14 - Apple. (2021). Retrieved from <https://www.apple.com/ios/ios-14/>
- Atavi. (2021). Retrieved from <https://atavi.com/>
- Raindrop.io — All-in-one bookmark manager. (2021). Retrieved from <https://raindrop.io/>
- Bookmark Ninja - Online Bookmark Manager. (2021). Retrieved from <https://www.bookmarkninja.com/>
- Pocket: How to Save. (2021). Retrieved from <https://getpocket.com/>
- Google - Bookmarks. (2021). Retrieved from <https://www.google.com/bookmarks/>
- Pinboard. (2021). Retrieved from <https://pinboard.in/>
- New Tab Redirect. (2021). Retrieved from <https://chrome.google.com/webstore/detail/new-tab-redirect/icpgifneehieebagbmdbhnlpio/pdcmna>

2. Requirements

2.1 Functional Requirements

Users can:

1. Create an account by logging in through Google account.
2. Create an account by signing up with email.
3. Log out or delete an account through account settings
4. Create and store bookmarks in folders.

5. Create folders for bookmarks.
6. Create and delete tasks using the Todo List widget.
7. Make, edit, style, and delete notes using the Notes widget.
8. * Add media in the notes.
9. Access weather information through the Weather widget.
10. Change preferred location for weather information in account settings
11. Customize background picture of the application
12. Position bookmarks in preferred grid order
13. Position widgets in preferred list order
14. Install an extension for Safari, Chrome, and Mozilla Firefox.
15. Create bookmarks by using the Safari, Chrome, or Mozilla Firefox extensions.
16. Set Command T as their start up page.
17. * Create an event using Calendar widget
18. * Share an event with the users' team using Calendar widget.
19. * Chat with users' team.
20. Get information about the installation and use of our web application.
21. Highlight and identify bookmarks by color tags.

System can:

1. Automatically save changes when the user uses Notes or Todo List widget.
2. Determine user's login status.
3. Save objects and their respective layout.
4. Permanently delete user's account

2.2 Use Cases

2.2.1 Use Case: Add a bookmark

Primary Actor:	User
Priority:	Essential
Precondition:	<ol style="list-style-type: none"> 1. User is logged in. 2. The user has copied the URL to add.
Trigger:	User finds a site he wants to bookmark.
Scenario:	<ol style="list-style-type: none"> 1. User selects to add a bookmark 2. System shows a screen to enter the URL, title, border color and folder location. 3. User pastes the URL, enters the title, chooses border color, and chooses folder. 4. User selects to save. 5. System verifies inputs are valid and displays updated bookmarks.
Extensions:	2-3a. User chooses to cancel editing. 2-3a.1 System removes the screen.

	5a. User left URL blank. 5a.1 System alerts to fill the URL field and proceed to Step 3.
--	---

2.2.2 Use Case: Customize layout of the objects

Primary Actor:	User
Priority:	Essential
Precondition:	User is logged in.
Trigger:	User wants to change the layout of the objects.
Scenario:	<ol style="list-style-type: none"> 1. User selects the setting. 2. System presents setting options (i.e. edit layout, edit profile, change background image). 3. User selects to edit layout. 4. System makes objects editable. 5. User changes the object's position as he desires. 6. User repeats step 5 as desired. 7. Users choose to cancel edit mode. 8. System disables objects from being editable.
Extensions	<p>2-3a. User cancels to select editing layout.</p> <p>2a. 1 System removes (or hides) setting options.</p> <p>5a. User tries to place an object outside of its section.</p> <p>5a. 1 The object returns to an original position.</p>

2.2.3 Use Case: Add a note using Notes widget

Primary Actor:	User
Priority:	Secondary
Precondition:	User is logged in.
Trigger:	User wants to make a note.
Scenario:	<ol style="list-style-type: none"> 1. User selects the Notes widget. 2. System displays the Note screen. 3. User selects to add a note. 4. System displays an empty note. 5. User enters a title and a content. 6. User repeats steps 3 to 6 as desired. 7. System automatically saves the note.
Extensions	<p>2-5a. User chooses to quit Notes.</p> <p>2-5a. 1 System automatically saves the note.</p>

	2-5a. 2 System removes the Notes screen. 6a. User did not enter a title and a content. 6a. 1 System does not autosave the note.
--	---

2.2.4 Use Case: Add a task using Todo List widget

Primary Actor:	User
Priority:	Secondary
Precondition:	User has logged in.
Trigger:	User wants to add a task to the Todo List.
Scenario:	<ol style="list-style-type: none"> 1. User selects the Todo List widget. 2. System displays the Todo List screen. 3. User enters his or her task to add to a todo list. 4. Users repeat step 3 as desired.
Extensions	<ol style="list-style-type: none"> 3a. User chooses to add empty task <ol style="list-style-type: none"> 3.a 1 System does not save an empty task. 4a. User is done with adding tasks. <ol style="list-style-type: none"> 4.a. 1 System autosaves the tasks. 4.a. 2 System quits the Todo List screen.

2.2.5 Use Case: Move a bookmark into another folder.

Primary Actor:	User
Priority:	Essential
Precondition:	User is logged in and has at least one bookmark in a folder.
Trigger:	User wants to move a bookmark into another folder.
Scenario:	<ol style="list-style-type: none"> 1. User selects settings. 2. System presents the settings options. 3. User selects the “Edit layout” option. 4. System allows objects to be editable. 5. User drags a bookmark on top of a folder he wants to move it to. 6. System moves the location of the bookmark into the folder and shows updated bookmarks.
Extensions	<ol style="list-style-type: none"> 3a. User drags a bookmark on top of the current folder. <ol style="list-style-type: none"> 3a.1. System restricts such action.

2.2.6 Use Case: Delete Objects

Primary Actor:	User
Priority:	Essential
Precondition:	User has logged in.
Trigger:	User wants to remove an object.
Scenario:	<ol style="list-style-type: none"> 1. User goes to Settings and chooses to change the current layout. 2. System allows objects to be editable. 3. User selects an object he wants to remove. 4. System displays a screen with an alert that tells the user to confirm his action. 5. User confirms to remove the object. 6. System removes the object.
Extensions	5a. User cancels deletion. 5a.1. System closes the screen.

2.2.7 Use Case: Add bookmark using the extension

Primary Actor:	User
Priority:	Essential
Precondition:	<ol style="list-style-type: none"> 1. User is logged in. 2. User installed the necessary extension. 3. User is on a different webpage (other than Command T).
Trigger:	User wants to bookmark the webpage he or she is visiting.
Scenario:	<ol style="list-style-type: none"> 1. User clicks the extension icon. 2. System shows a screen with the current page's URL, title, default folder, and default color (default folder is first folder and default color is clear color). 3. User chooses the folder location, color tag, and saves the bookmark. 4. System verifies the URL and title is valid. 5. System saves the bookmark into the folder.
Extensions	2a. User wants to add the bookmark into a new folder 2a.1 System restricts such actions. 4a. Empty URL and title 4a.1 System cancels the adding bookmark process.

2.2.8 Use Case: Change the background image

Primary Actor:	User
Priority:	Essential
Precondition:	User is logged in.
Trigger:	User wants to change the background image.
Scenario:	<ol style="list-style-type: none"> 1. User selects the setting. 2. System presents setting options (i.e. edit layout, edit profile, change background image). 3. User chooses to change the background image. 4. System shows a screen with photos and allows the user to search more on the internet. 5. User selects a photo. 6. System updates the change.
Extensions	3-5a. User cancels to change background image 3-5a.1 System closes the screen with images and shows Command T.

2.3 User Interfaces

[Link to our Figma Prototype](#)

1. Landing page

 Command T

Change the way you organize your bookmarks and use your homepage

Get started by logging in with your account or sign up with your Google account or email.

Already have an account? Log in with your email below.

[Log in](#)

New here? Sign in with Google or create a new account.

[Sign in with Google](#)

[Create an account](#)

Scroll down to discover more about Command T ▾



This is the first screen that people who are not logged in to Command T will see, when they visit Command T. Existing users can select “Log in”, and new users can select to “Sign in with Google” or “Create an account”.

1.1 Login Page

⌘ Command T



Log in

Already have an account? Log in with your email below.

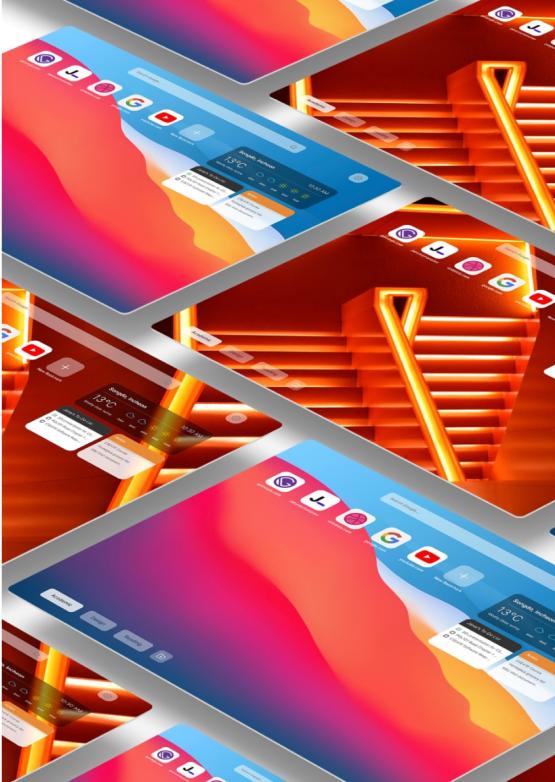
Email

Password

Log in

1.2 Create a new account page

⌘ Command T



Create an account

Enter your information to create a new account.

Full name

Username

Email

Password

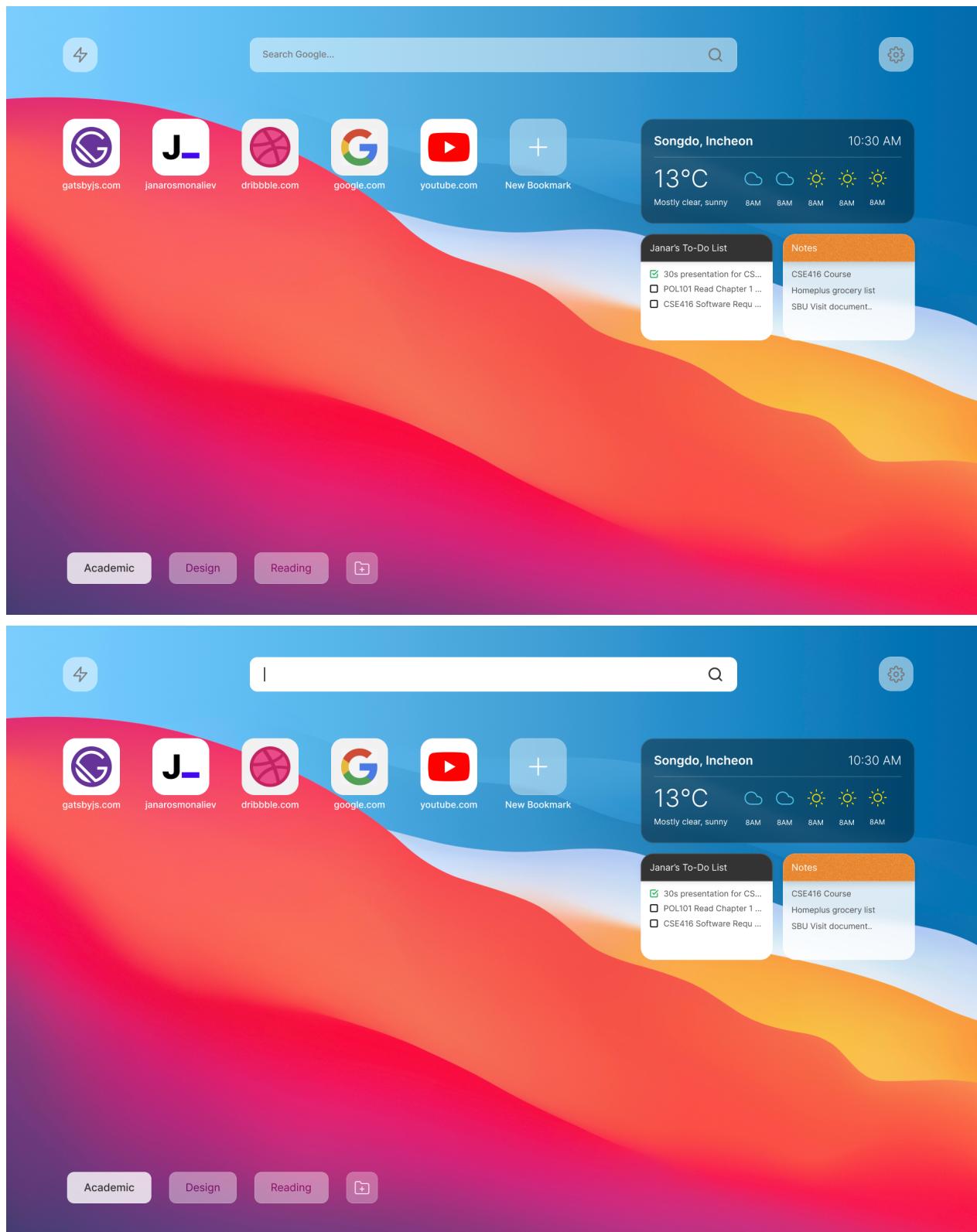
Current city

Create an account

[Privacy Policy](#)

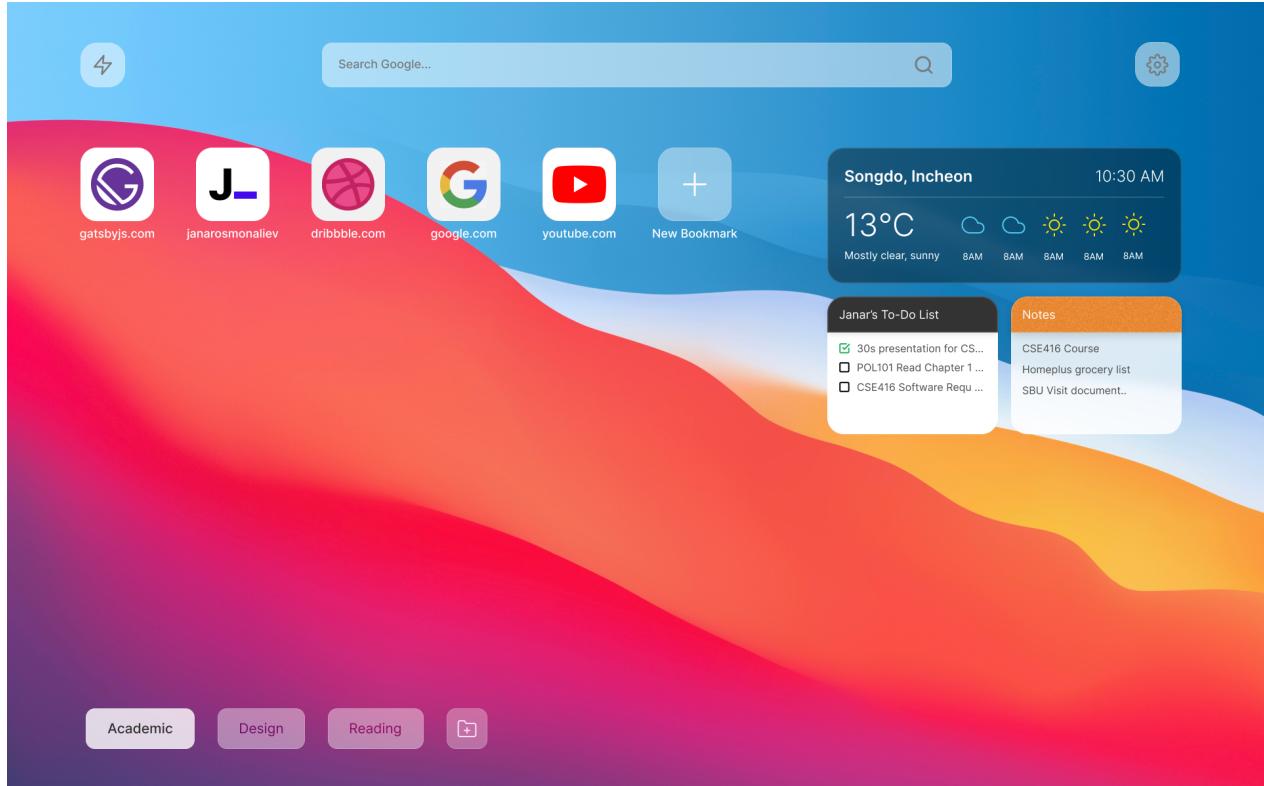
2. App homepage

2.1 Use Search bar



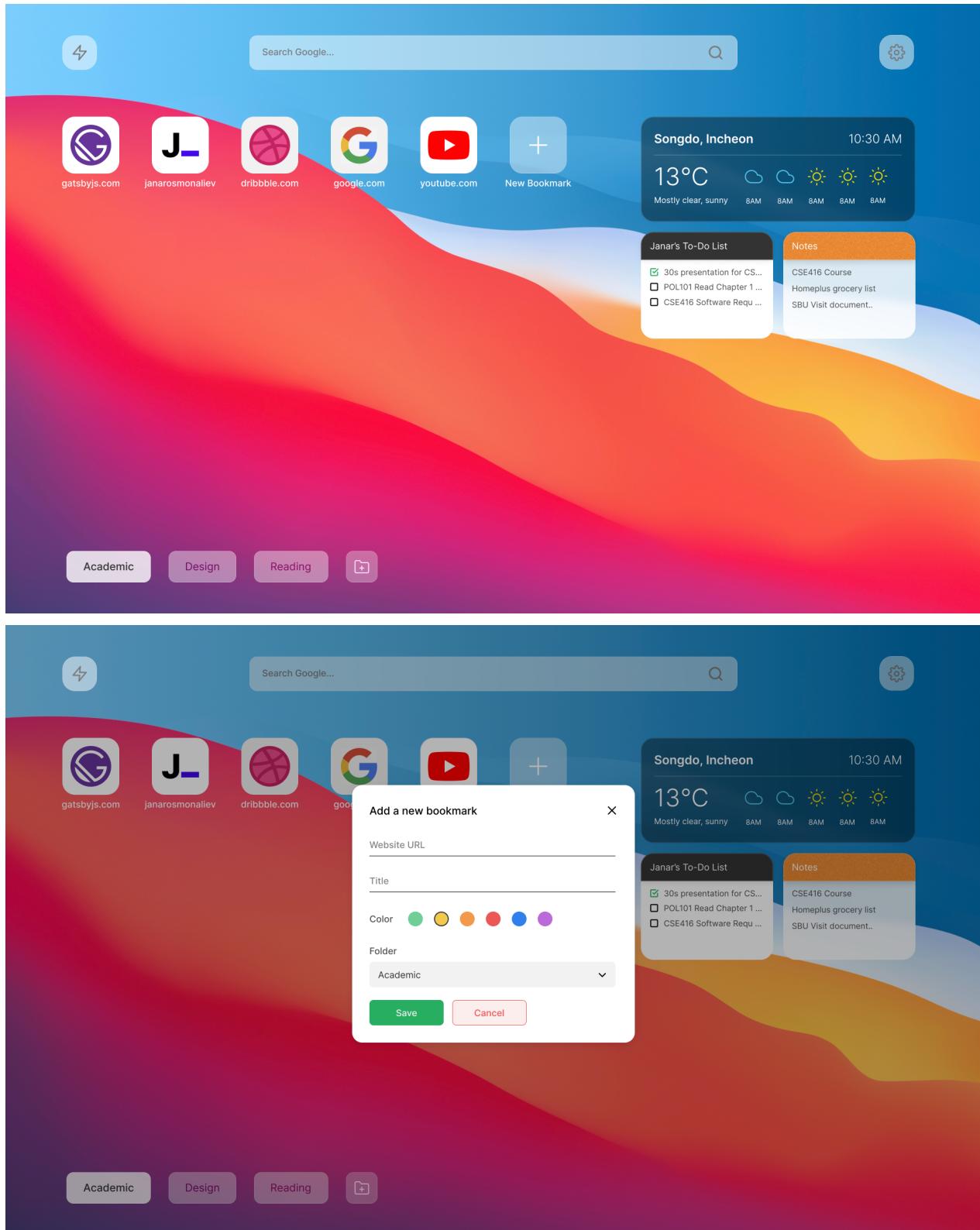
Users can search something on Google by clicking the search bar.

2.2 Switching between folders



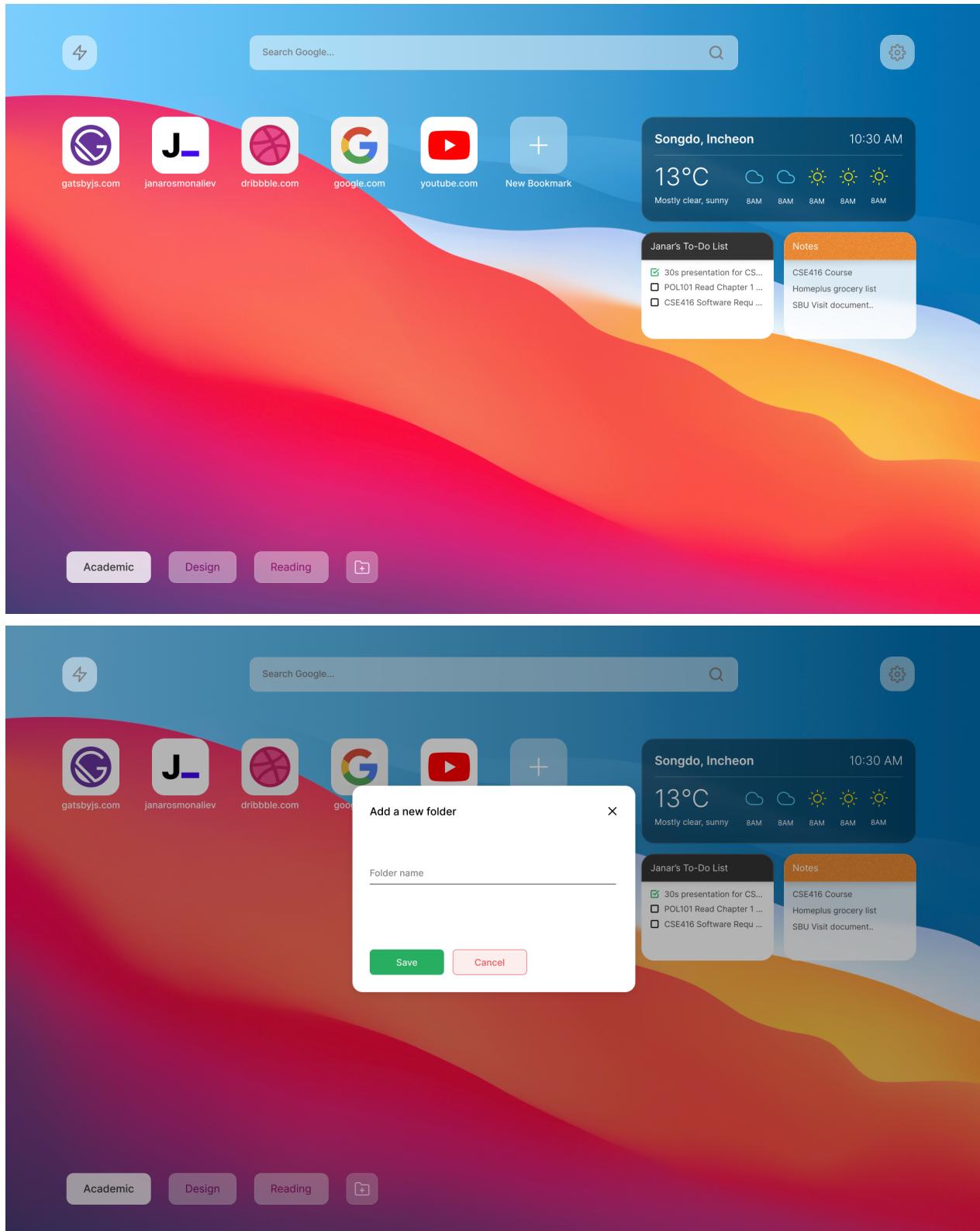
Users can see bookmarks in different folders by selecting a folder.
Bookmarks from the “Reading” folder will appear immediately after clicking.

2.3 Adding a new bookmark



Covers “Add a bookmark” use case.
Users can click the “New Bookmark” button.

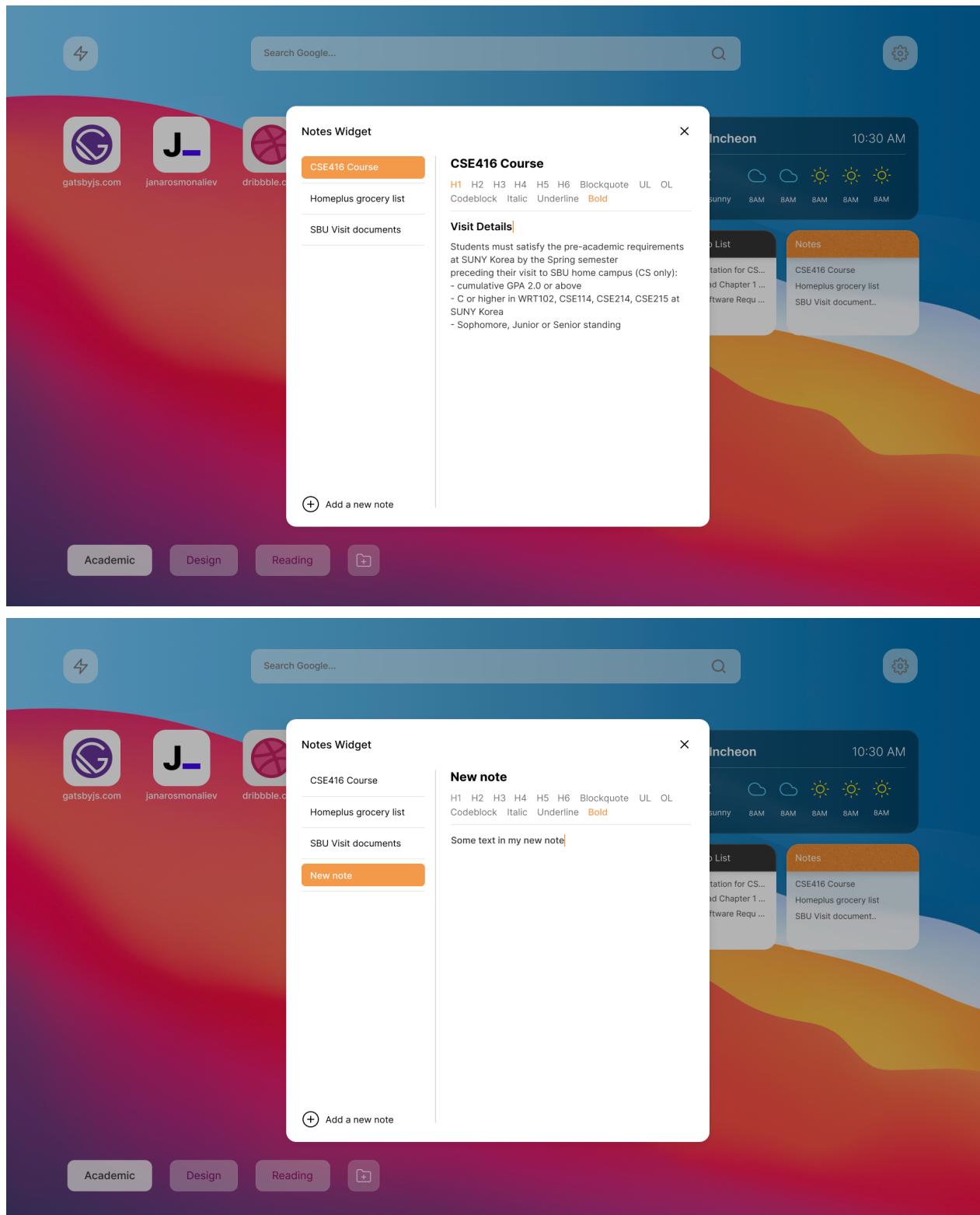
2.4 Adding a new folder



Covers “Move a bookmark into another folder” use case.
Users can click the “New Folder” button.

3. Widgets

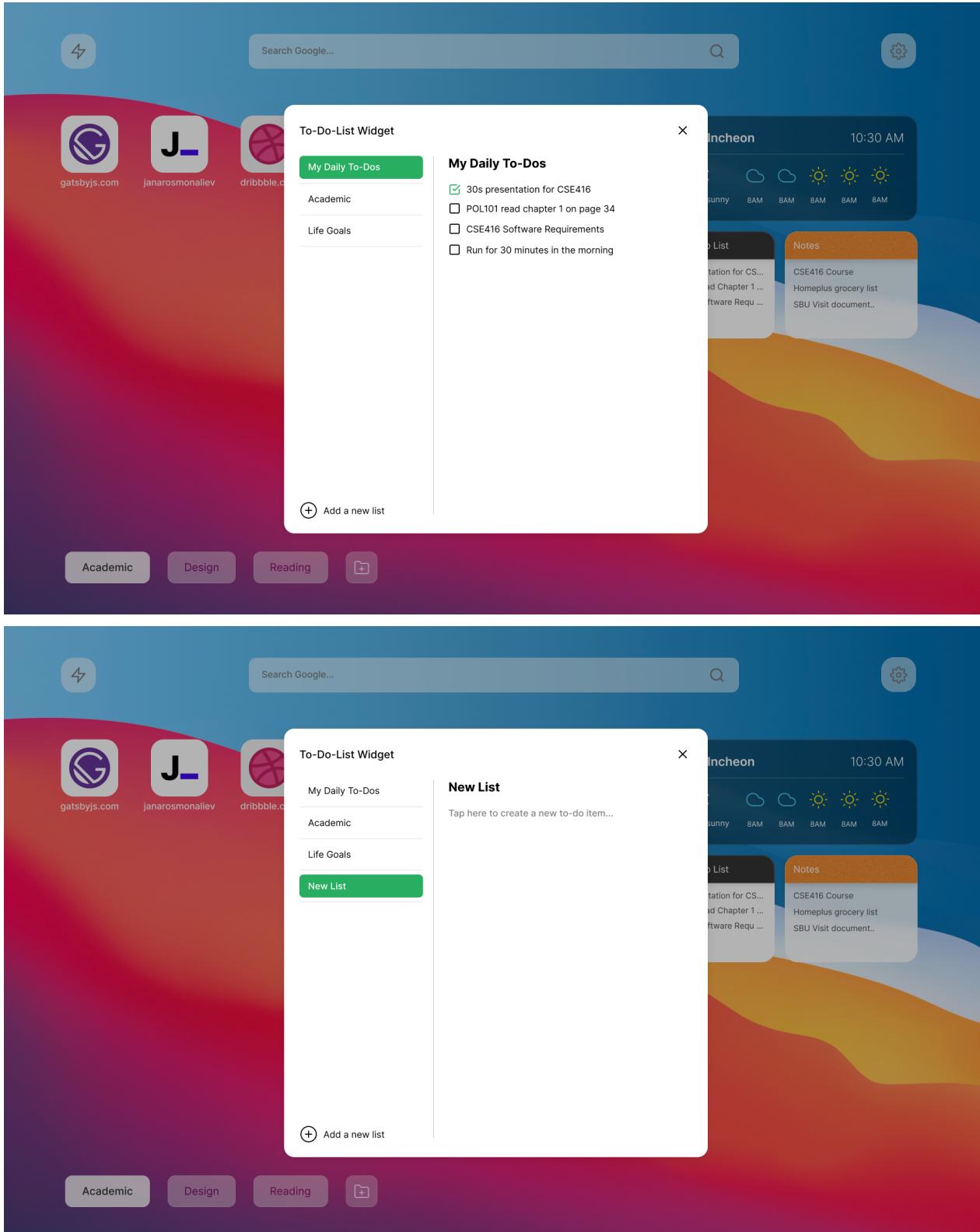
3.1 Notes Widget



Users can click the Notes widget to add or edit their notes.

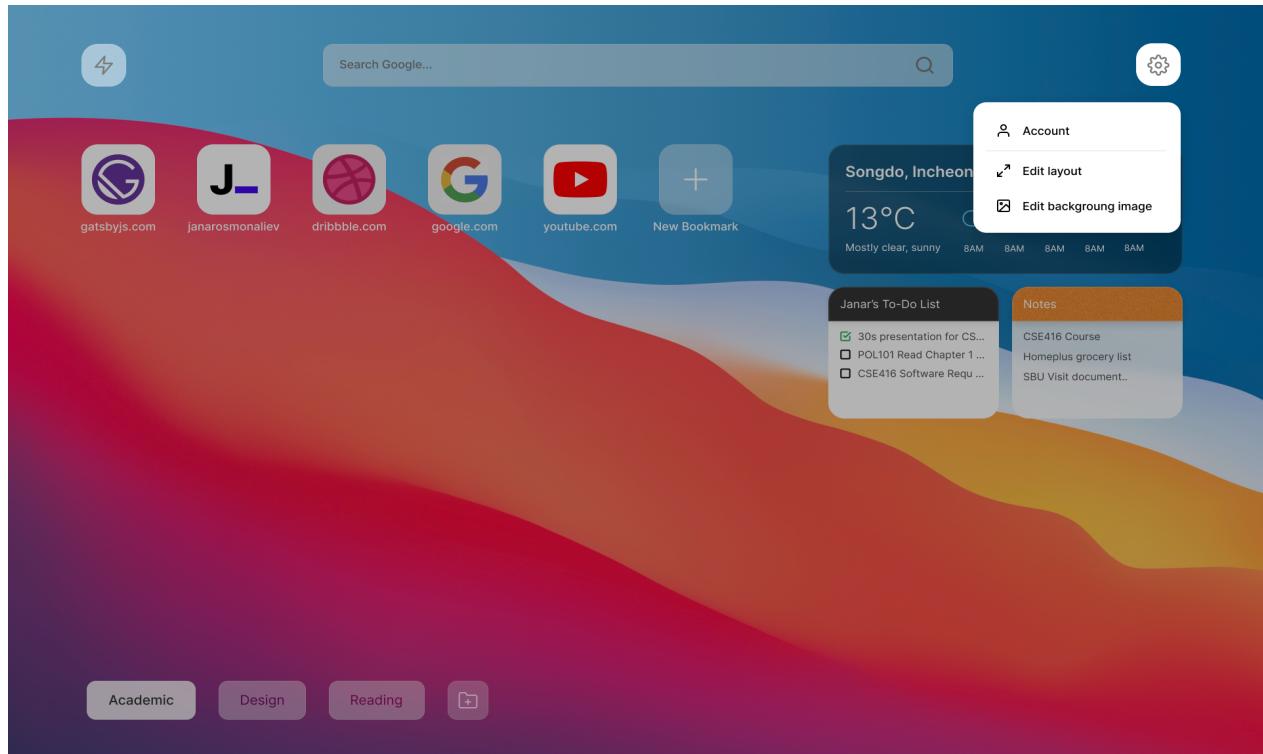
Covers “Add a note using Notes widget” use case

3.2 Todo List Widget

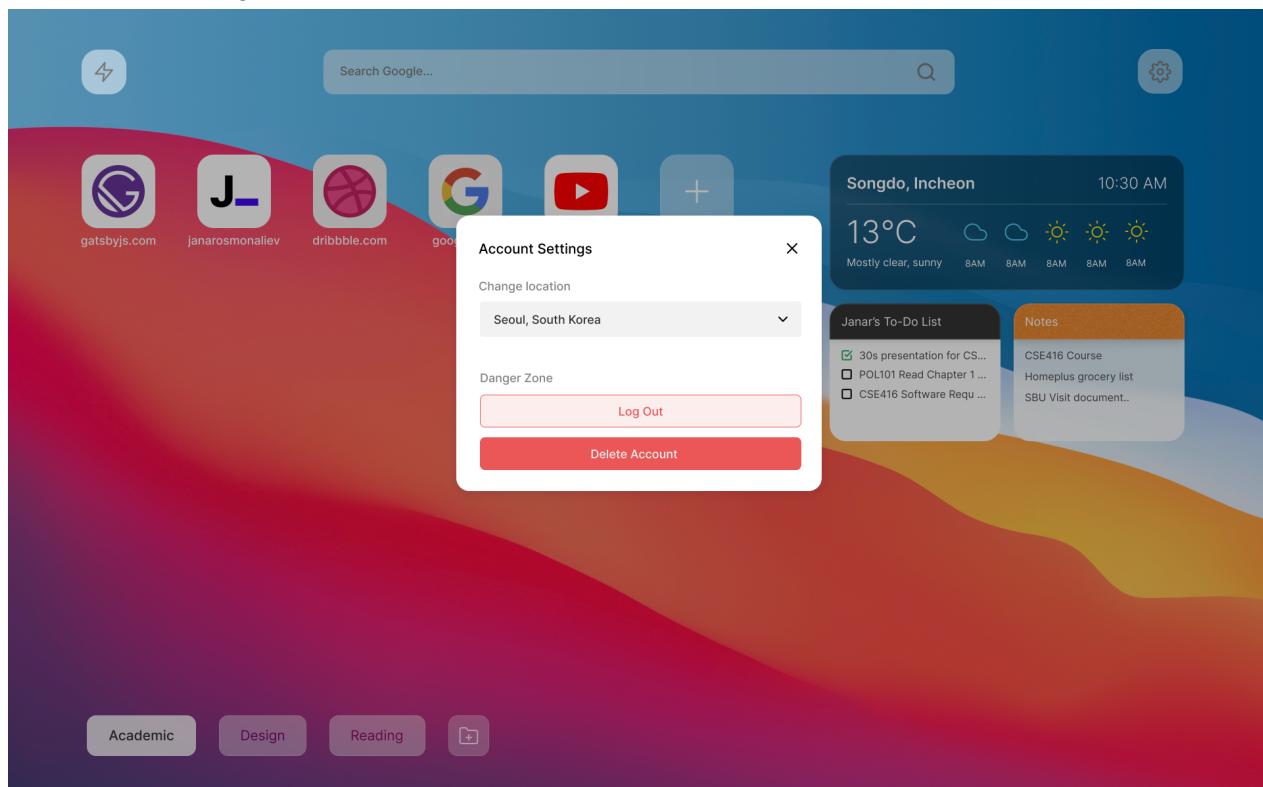


Users can click the Todo List widget to add tasks.
Covers “Add a task using Todo List widget” use case

4. Settings

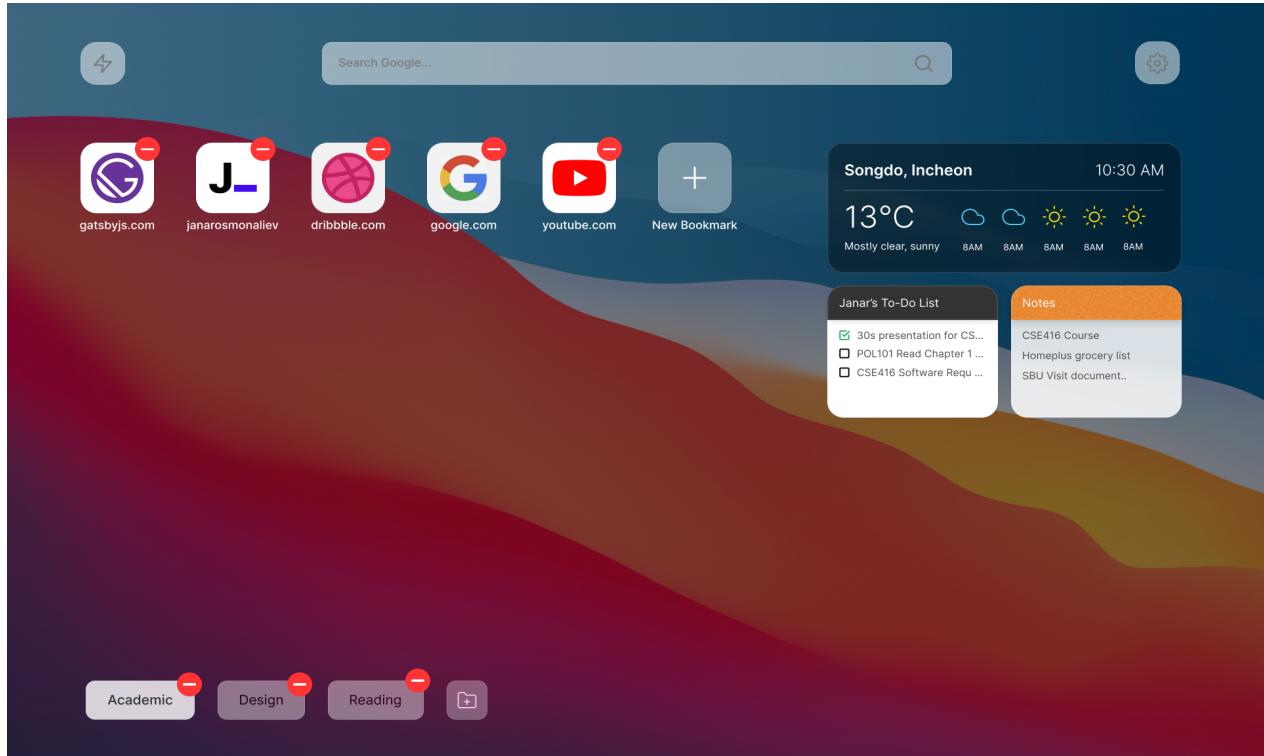


4.1 Account settings

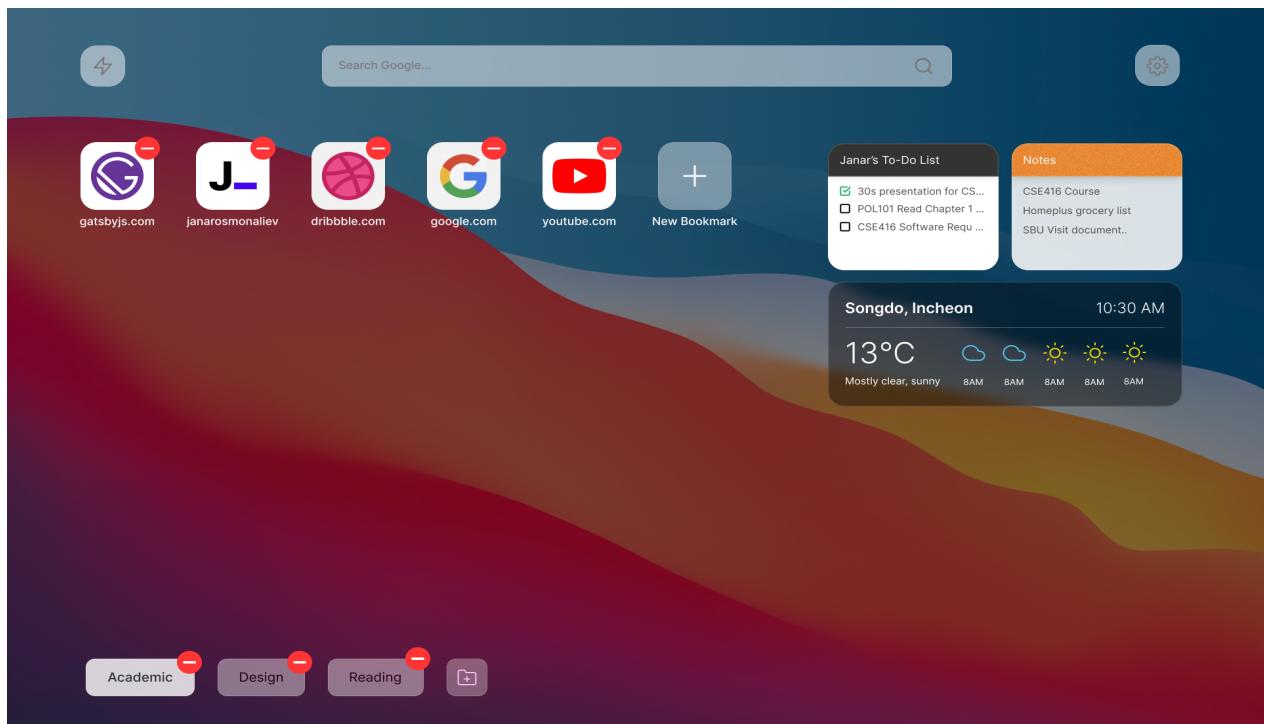


Account settings modal

4.2 Edit layout

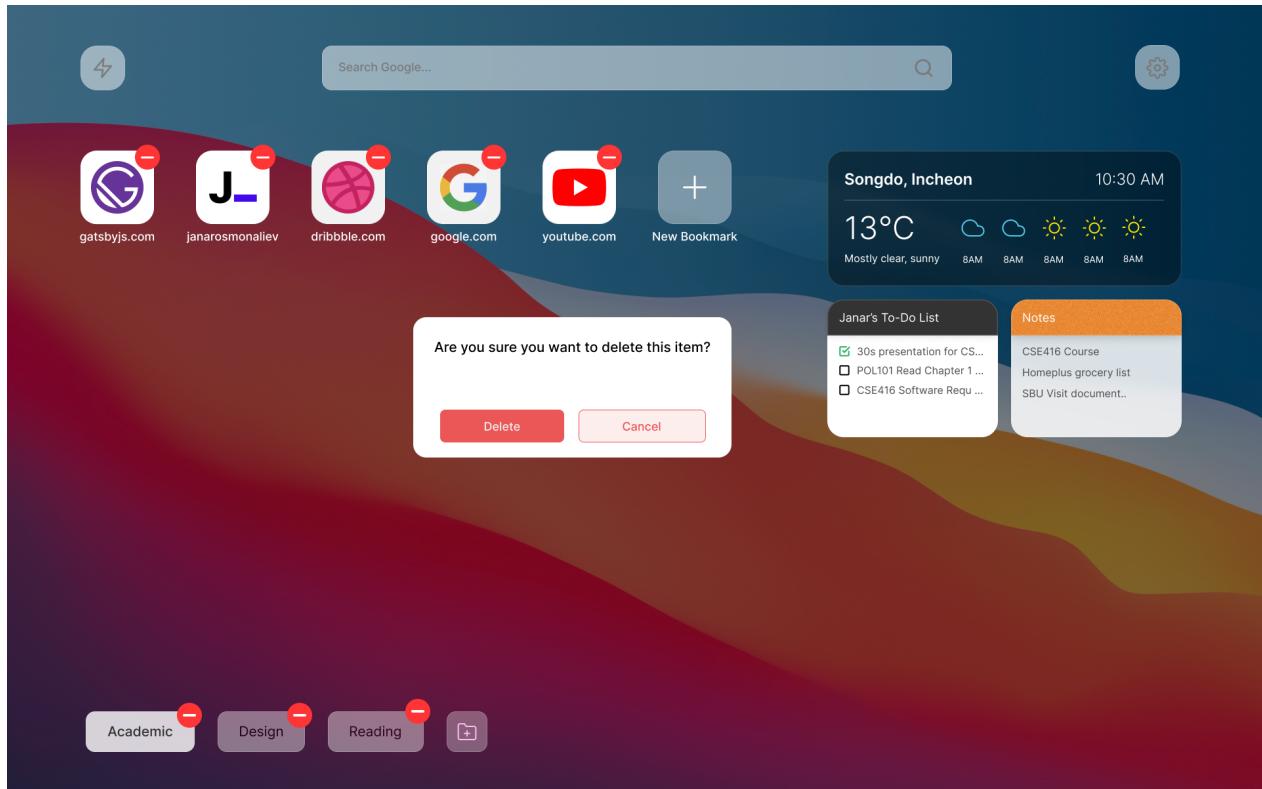


4.2.1. Change widget layout



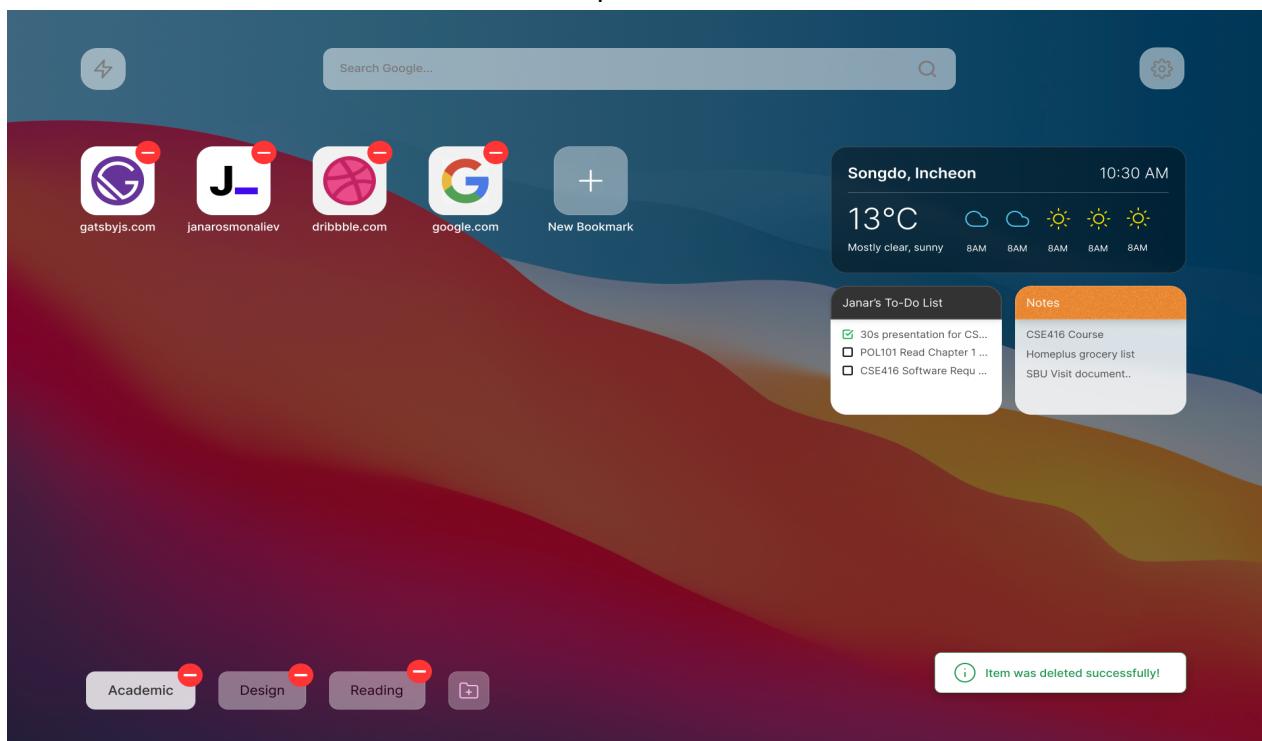
Covers “Customize layout of the objects” use case. The user selects the edit layout option from the settings menu. All the objects display a minus sign on the top-right corner and allow the user to reposition them on the screen.

4.2.2. Delete a bookmark



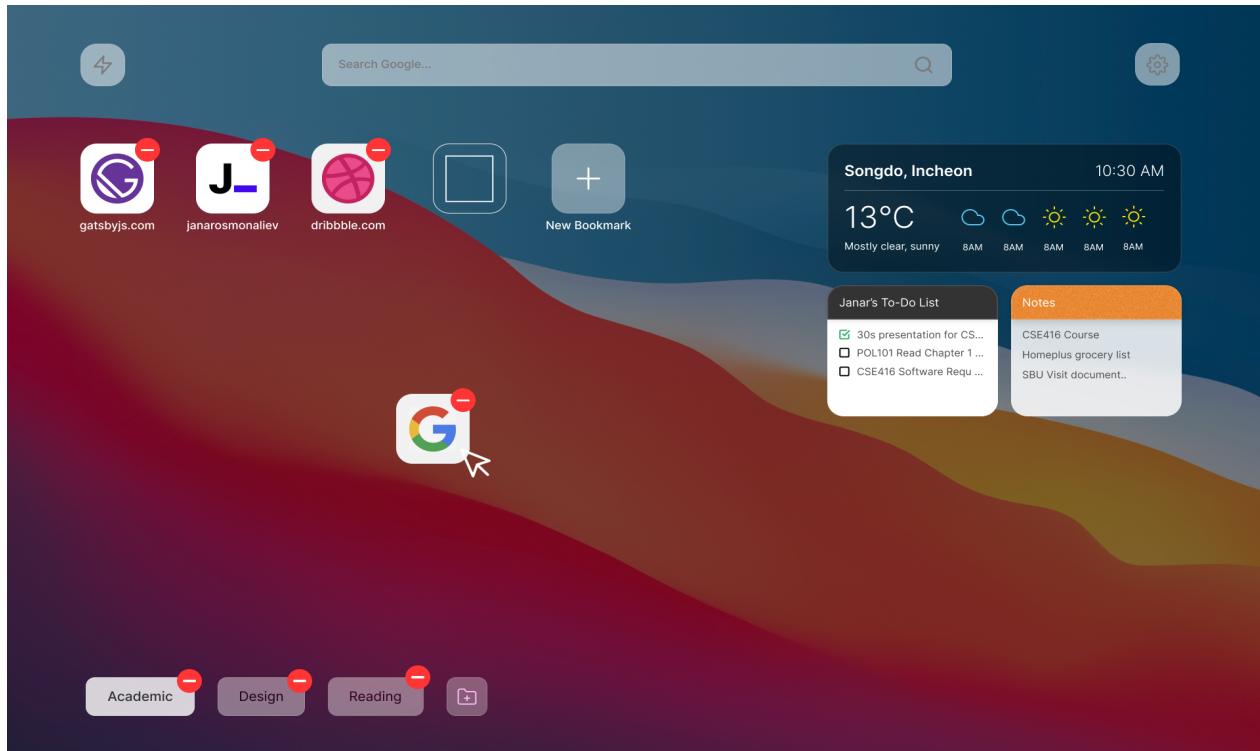
A user can click the delete bookmark icon that he or she wants to remove.

A user can choose to proceed or cancel deletion.

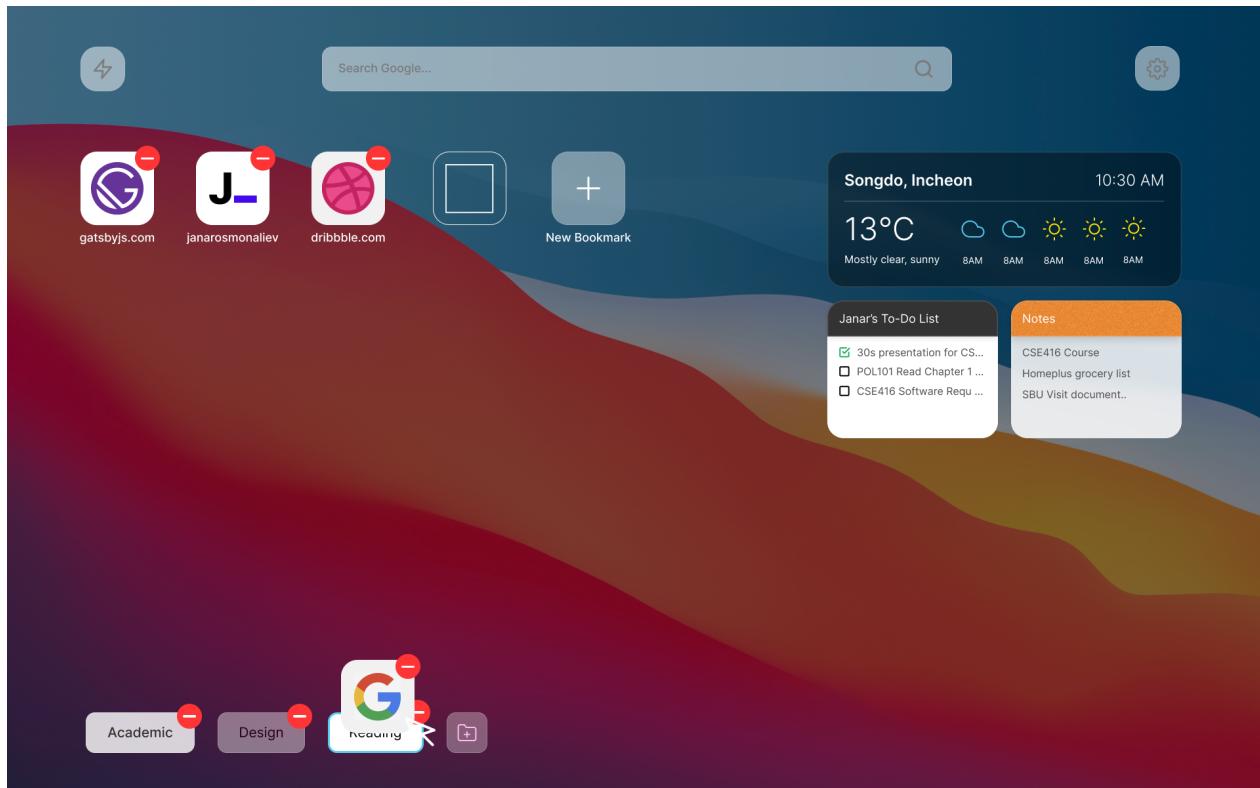


Covers “Delete Objects” use case. The user clicks on the minus sign button and a prompt asks to confirm the delete action. Upon confirmation, the bookmark is eliminated from the screen.

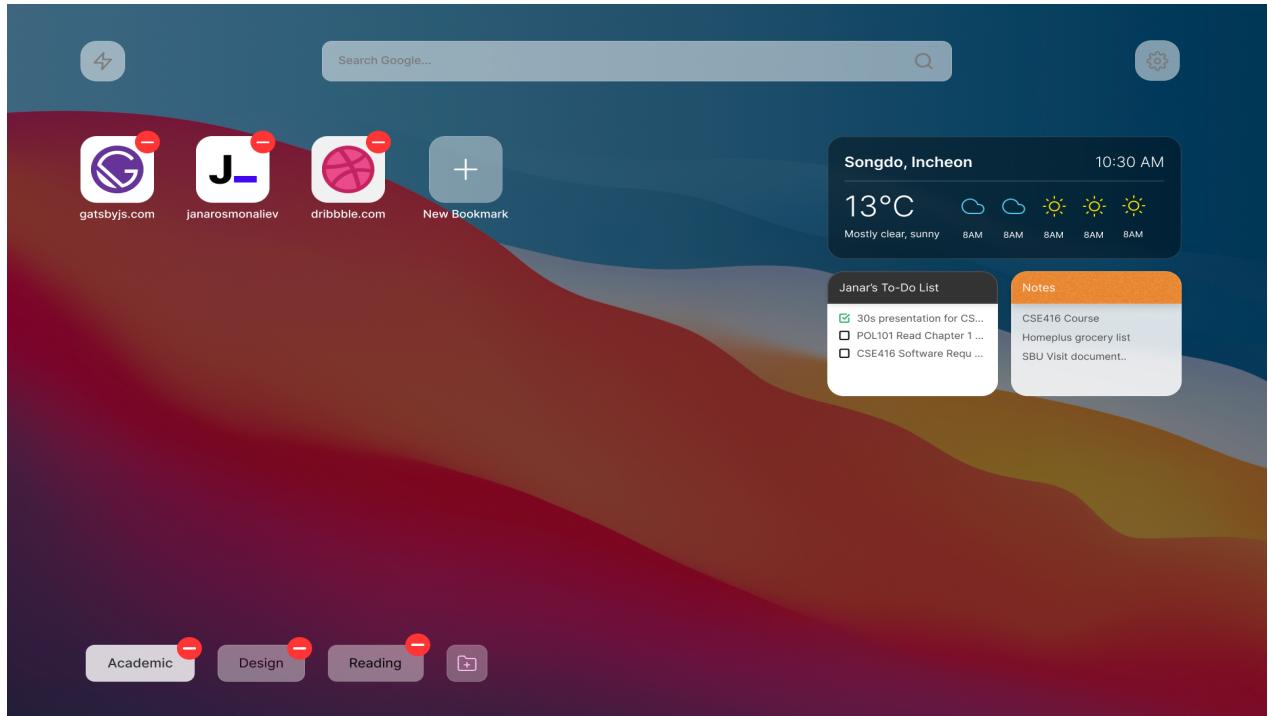
4.2.3. Move the bookmark into different folder



Covers “Move a bookmark into another folder” use case. The user can drag and drop a bookmark into one of the available folders.

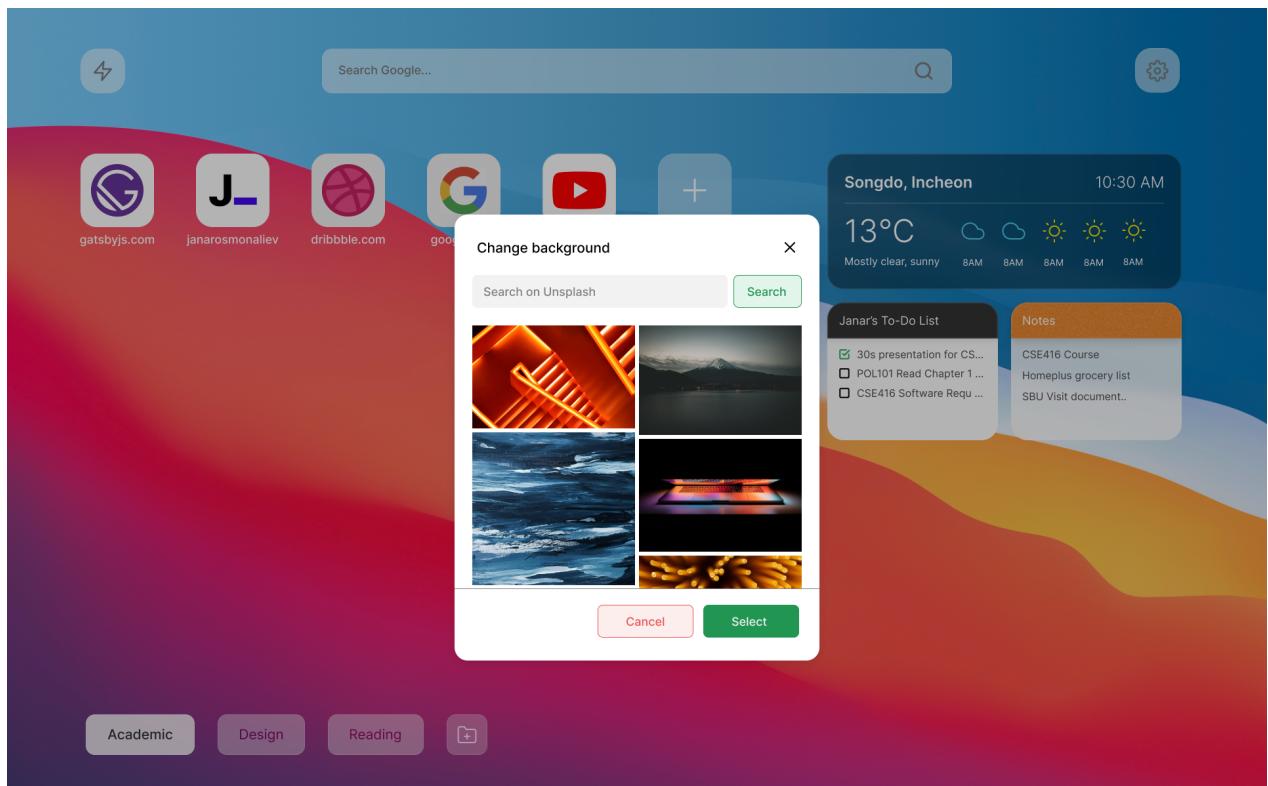


If user hovers the bookmark above any folder after dragging it there, folder will change color and indicate that the bookmark can be dropped into it.

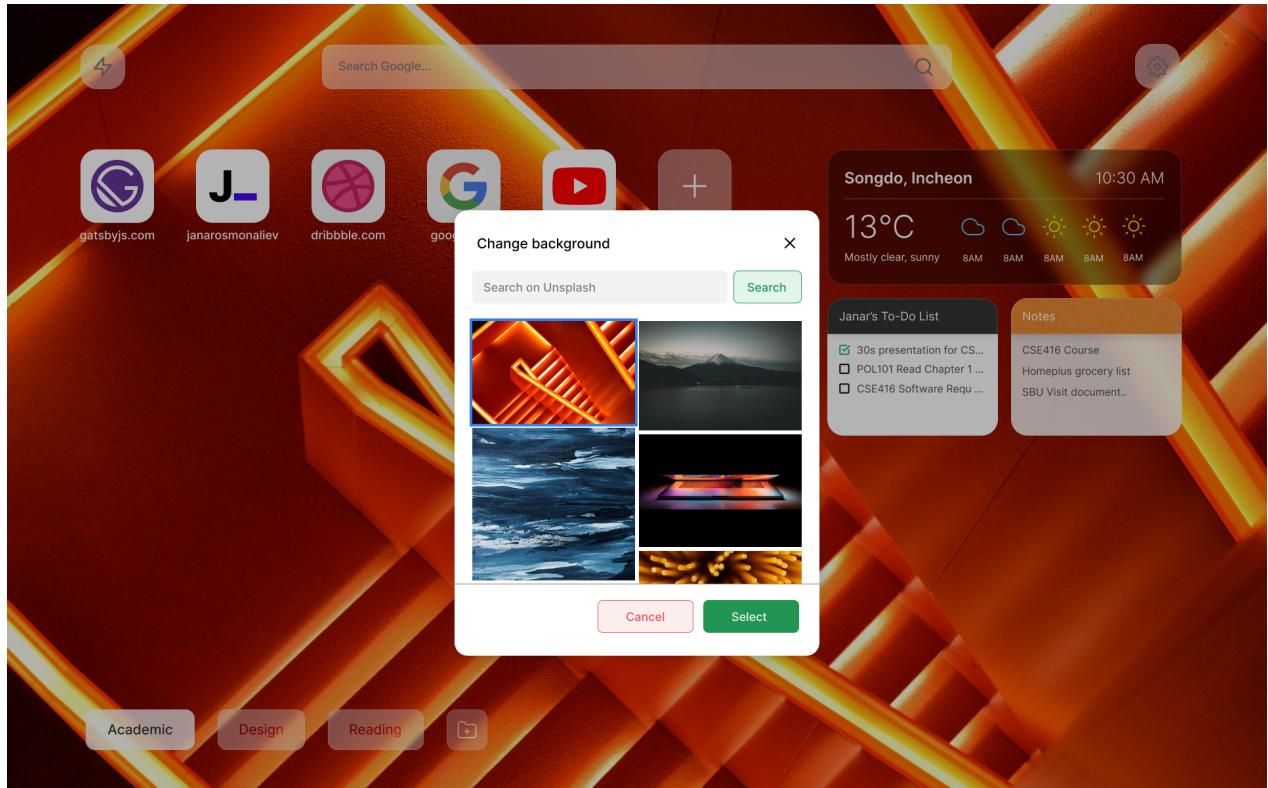


The bookmark is successfully moved to another folder

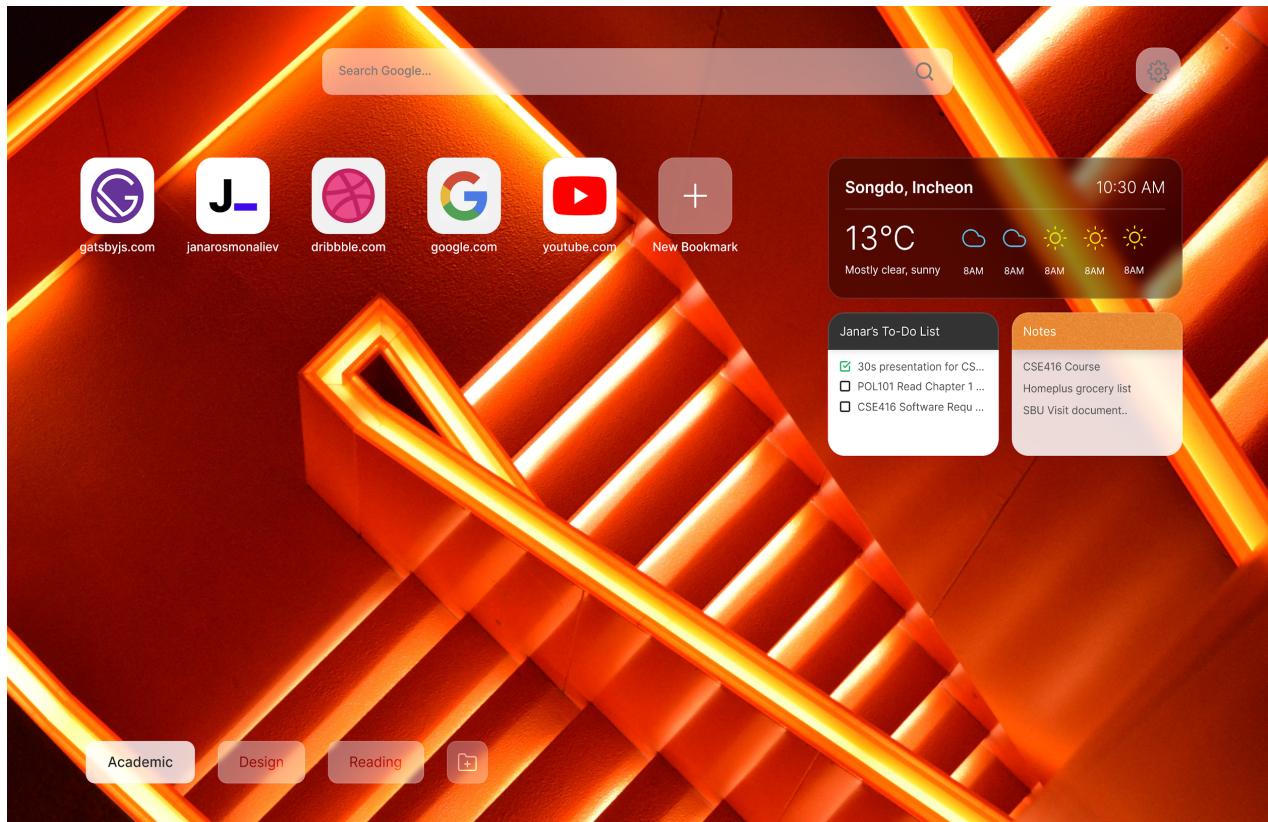
4.3 Edit background picture



Covers “Change the background image” use case. A screen displays the available images.

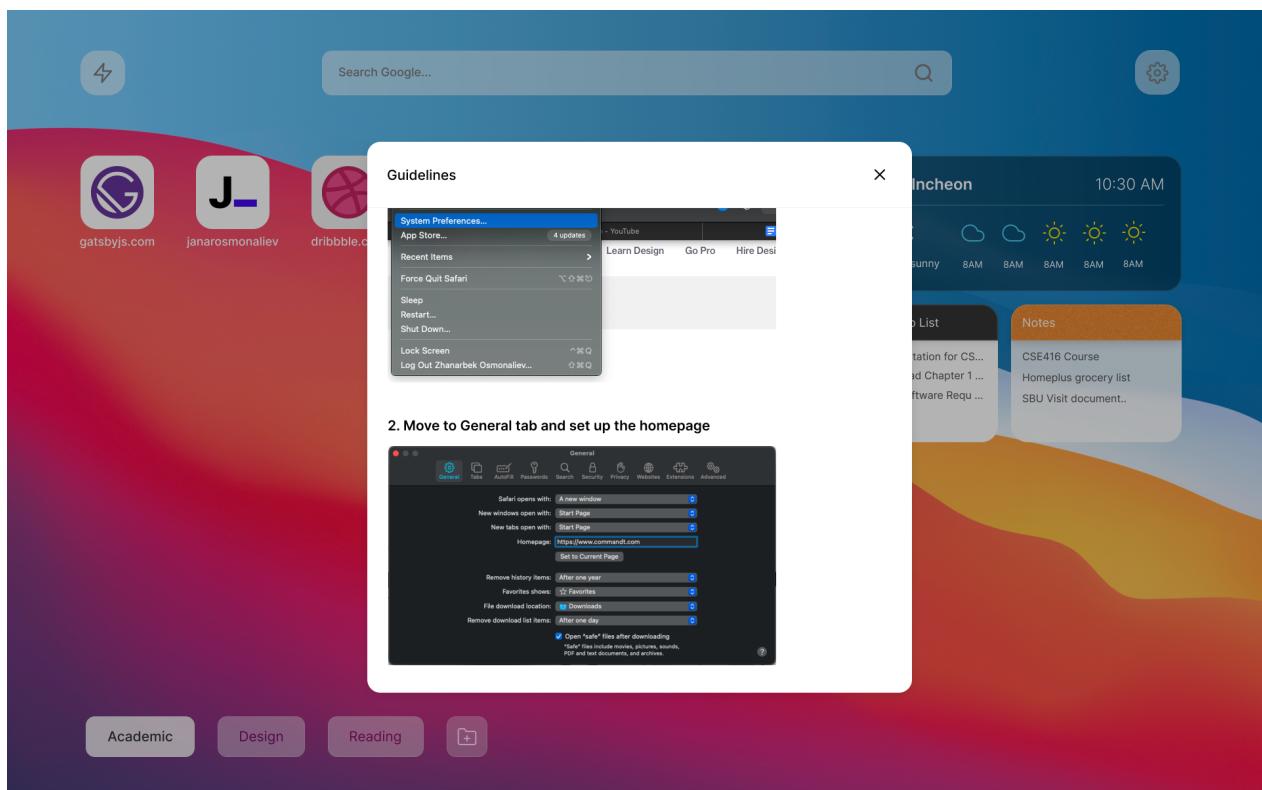
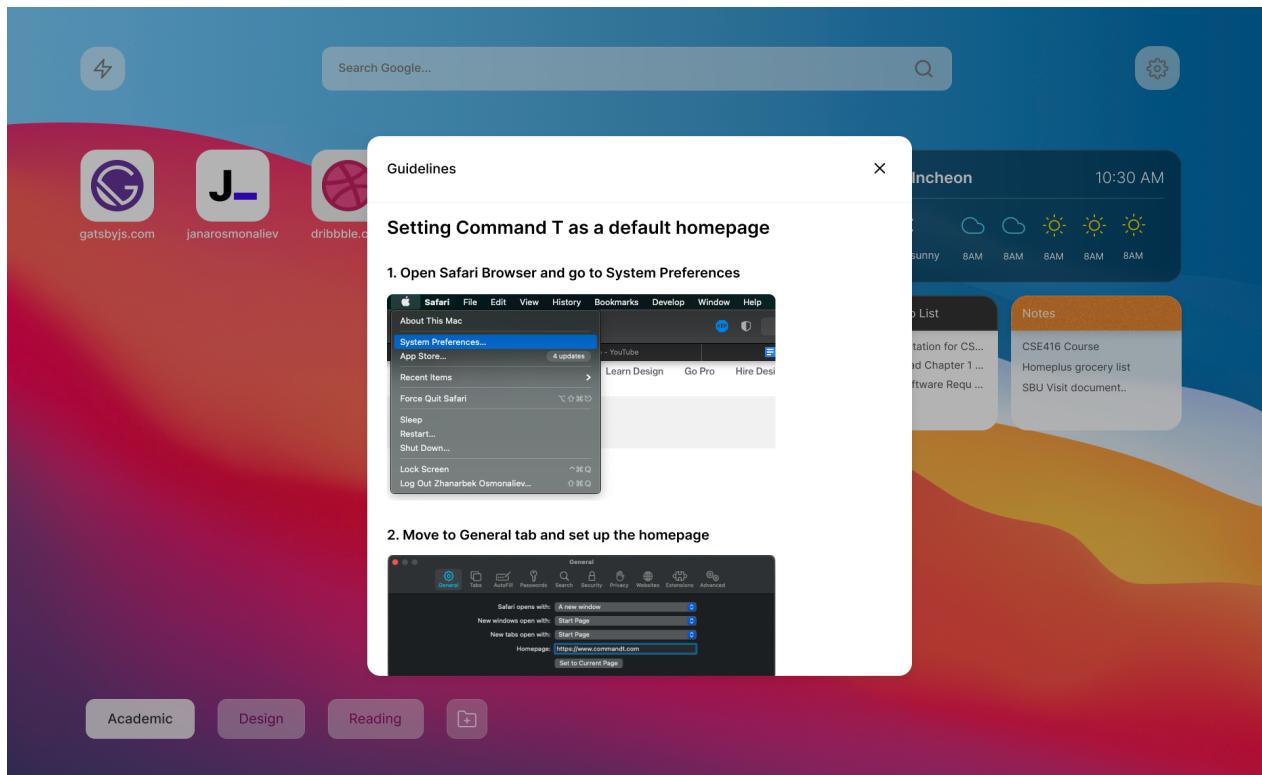


Clicking on an image updates the background and the select button saves the changes.



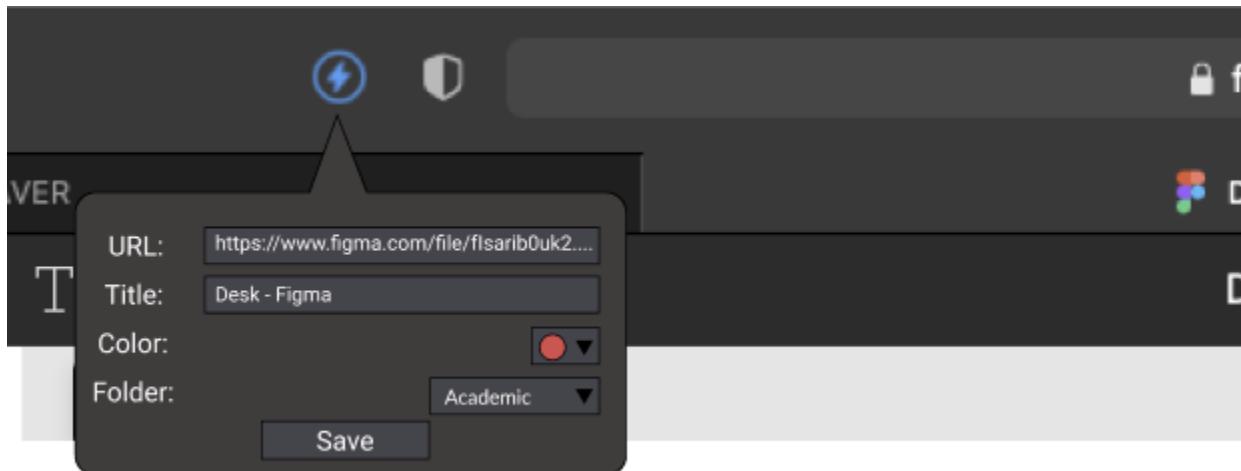
Command T screen with the new background image.

5. Guidelines



Users can refer to the guidelines to install an extension as well as to set the application as the default browser homepage.

6. Use Extension



Covers “Add bookmark using the extension” use case. A user clicks an extension icon to save the bookmark and the system presents a screen with current URL and title.

2.4 Non-functional Requirements

- Users can use all functions in Google Chrome, Safari, and Firefox on desktop web browsers.
- Users’ data will be stored safely with Google’s OAuth 2.0.
- Users’ data will not be visible to other users but only to system administrators.
- Page will be loaded under 500 ms.
- The error in all functions provided on our website is less than 1 percent.

3. Contributions

Fabio Calero:

- Wrote Product Description, Scope, Users, Existing Alternatives, Definitions, and References.
- Revised Software Requirement Specification.
- Added annotations for UI Designs.

Hasung Jun:

- Got feedback from Zero Class Lab and professor Khun
- Revised Use Cases #1~4
- Wrote Non-functional Requirements
- Revised Software Requirement Specification.
- Added annotations for UI Designs.

Zhanarbek Osmonaliev:

- Interdisciplinary project pitch on March 16th.
- Presentation slides for pitch on March 16th.
- Made the Figma prototype and UI design.
- Wrote Functional Requirements.

Yejin Shin:

- Group project pitch on March 4th
- Wrote about future user feedback for User Feedback.
- Wrote the draft of Use Cases and revised Use Cases #5~8.
- Revised Software Requirement Specification.
- Added annotations for UI Designs.