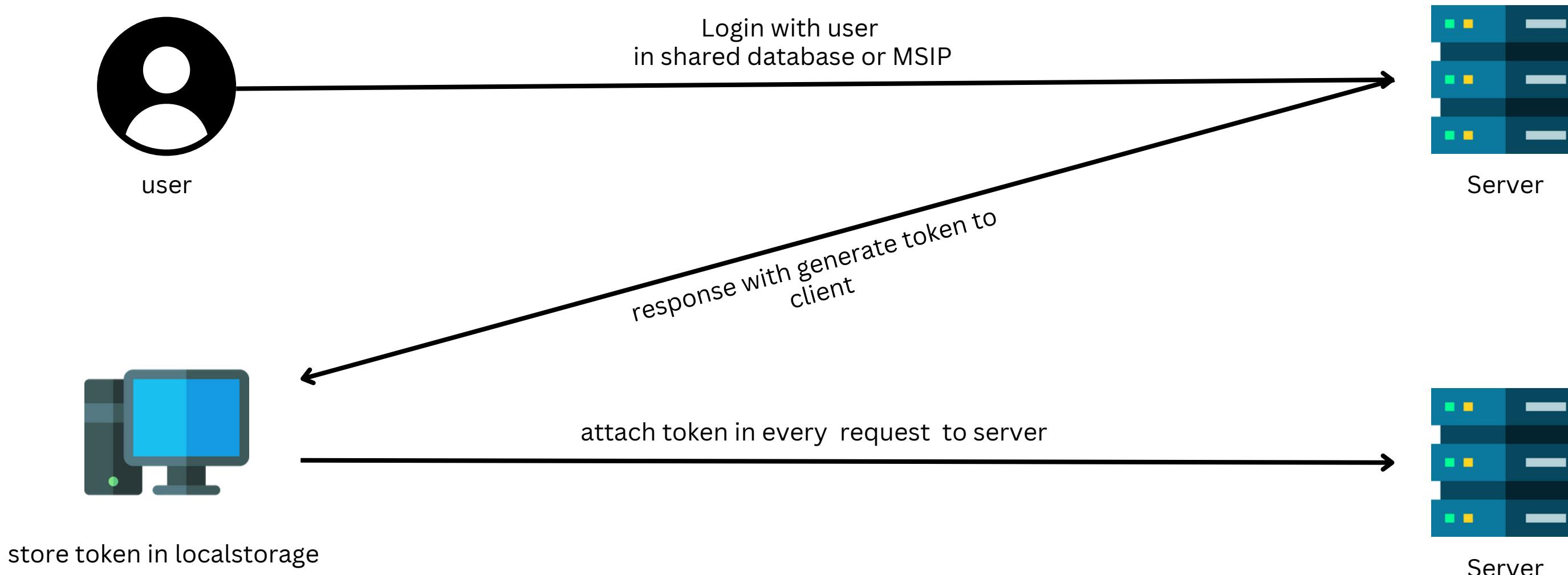


FRONTEND

AUTHENTICATION/TOKEN



Login with shared user

Login function

```
async function signIn(user) {
  try {
    toastStore.displayLoading()
    const response = await fetch(`${url}/login`, {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify(user),
    })
    await toastStore.resetToast()

    if (response.ok) {
      const token = await response.json()
      const userTokenObject = authStore.addToken(token.access_token, token.refresh_token, 'local')
      toastStore.changeToast('success', 'Success', 'You have successfully logged in')

      return userTokenObject
    } else if (response.status === 400 || response.status === 401) {
      toastStore.changeToast('error', 'Error', 'Username or Password is incorrect')
    } else {
      toastStore.changeToast('error', 'Error', 'There is a problem. Please try again later')
    }
  } catch (error) {
    console.error(`Error during sign in: ${error}`)
    throw error
  }
}
```

addToken

```
const addToken = (newAccessToken, newRefreshToken, newType) => {
  accessToken.value = newAccessToken
  refreshToken.value = newRefreshToken || refreshToken.value || ''
  typeLogin.value = newType || typeLogin.value
  let userTokenObject

  userTokenObject = {
    username: decodeToken(newAccessToken).username,
    access_token: accessToken.value,
    refresh_token: refreshToken.value,
    typeLogin: typeLogin.value,
  }

  localStorage.setItem('authData', JSON.stringify(userTokenObject))
  isLogin.value = true
  return userTokenObject
}
```

ฟังก์ชัน login ใช้สำหรับการยิง request ไปที่ server เพื่อกำ authentication เมื่อ response status เป็น ok จะได้รับ generate token มาจาก server และจะมีการเรียกใช้ฟังก์ชันในการที่จะเก็บ Token เข้าไปใน LocalStorage

ฟังก์ชันที่ใช้สำหรับการ addToken ไปเก็บไว้ใน LocalStorage

Login with shared user

getToken

```
const getToken = () => {
  const auth = JSON.parse(localStorage.getItem('authData'))
  isLogin.value = auth ? true : false
  typeLogin.value = auth ? auth.typeLogin : ''
  accessToken.value = auth ? auth.access_token : null
  refreshToken.value = auth ? auth.refresh_token : null
  return accessToken.value
}
```

fetchWithToken

```
async function fetchWithToken(endpoint, options = {}) {
  if (count > 1) return

  await authStore.checkToken()
  const token = authStore.getToken()

  const headers = {
    'Content-Type': 'application/json',
    ...options.headers,
  }

  if (token) {
    headers['Authorization'] = `Bearer ${token}`
    const loginType = await authStore.getTypeOfLogin()
    headers['Auth-Type'] = loginType.toUpperCase()
  }

  toastStore.displayLoading()

  let response = await fetch(`${url}${endpoint}`, {
    ...options,
    headers,
  })
  await toastStore.resetToast()

  if (response.status == 401) {
    count++
    await authStore.refreshAccessToken()
    if (authStore.getToken()) {
      headers['Authorization'] = `Bearer ${authStore.getToken()}`
      response = await fetch(`${url}${endpoint}`, {
        ...options,
        headers,
      })
    } else count = 0
    return response
  }
}
```

ใช้ในการดึงข้อมูลจาก authData (token) ที่เก็บเอาไว้ใน Local Storage

เป็นฟังก์ชันที่ทำการ config หรือปรับแต่งตัว fetch api เอาไว้ให้แบบตัว token ไว้ใน header ซึ่งเมื่อเราจะทำการส์ request ต่างๆ

สามารถส่งเข้าไปผ่านตัวฟังก์ชันนี้ได้ แบบตัวอย่างข้างล่าง

```
async function getAllBoard() {
  try {
    return (await fetchWithToken('/v3/boards')).json()
  } catch (error) {
    console.error(`Error fetching boards: ${error}`)
  }
}
```

Login with ms ip

LoadAzureData

```
const loadAzureData = async () => {
  try {
    await initializeMsal()

    const currentAccounts = msalInstance.getAllAccounts()
    if (currentAccounts.length === 0) {
      return null
    }

    const account = currentAccounts[0]
    msalInstance.setActiveAccount(account)

    try {
      const response = await msalInstance.acquireTokenSilent({
        ...loginRequest,
        account: account,
      })

      state.user = account
      state.isAuthenticated = true
      isLogin.value = true
      typeLogin.value = 'azure'

      const tokenKeys = JSON.parse(localStorage.getItem(`msal.token.keys.${import.meta.env.VITE_CLIENT_ID}`))
      const accessToken = JSON.parse(localStorage.getItem(`${tokenKeys.accessToken[0]}`))
      const refreshToken = JSON.parse(localStorage.getItem(`${tokenKeys.refreshToken[0]}`))
      const idKey = JSON.parse(localStorage.getItem(`${tokenKeys.idToken[0]}`))

      addToken(accessToken.secret, refreshToken.secret, 'AZURE')
      Edit Chat ...
    } catch (silentError) {
      console.error('Silent token acquisition failed:', silentError)
      if (silentError.name === 'InteractionRequiredAuthError') {
        return null
      }
      throw silentError
    }
  } catch (error) {
    console.error('Load Azure data error:', error)
    throw error
  }
}
```

azureLogin

```
const azureLogin = async () => {
  try {
    await initializeMsal()
    state.isAuthenticated = false
    state.user = null
    isLogin.value = false
    typeLogin.value = ''

    await msalInstance.loginRedirect(loginRequest)
    const response = await msalInstance.acquireTokenSilent({
      ...loginRequest,
      account: account,
    })

    await loadAzureData()
  } catch (error) {
    if (error.name === 'BrowserAuthError' &&
      error.message.includes('interaction_in_progress')) {
      localStorage.removeItem('msal.interaction.status')
      return
    }
    console.error('Azure login error:', error)
    throw error
  }
}
```

ใช้ในการโหลดข้อมูลก่อน ว่ามีการล็อกอินแล้วหรือยัง เนื่องจาก
อาจจะมี Token อญ্য

ฟังก์ชันในการล็อกอินผ่านตัว msal ที่เป็น dependency
ของฟัง front-end

Login with ms ip

azureHandleRedirect

```
const azureHandleRedirect = async () => {
  try {
    await initializeMsal()

    const response = await msalInstance.handleRedirectPromise()

    if (response) {
      const account = response.account
      msalInstance.setActiveAccount(account)

      // Store user info
      state.user = account
      state.isAuthenticated = true
      isLogin.value = true
      typeLogin.value = 'azure'

      return account
    } else {
      return await loadAzureData()
    }
  } catch (error) {
    console.error('Handle redirect error:', error)
    if (error.name === 'BrowserAuthError' && error.message.includes('interaction_in_progress')) {
      localStorage.removeItem('msal.interaction.status')
      return await loadAzureData()
    }
    throw error
  }
}
```

getGraphToken

```
const getGraphToken = async () => {
  try {
    await initializeMsal()

    const account = msalInstance.getActiveAccount()
    if (!account) {
      throw new Error('No active account')
    }

    const response = await msalInstance.acquireTokenSilent({
      ...graphScopes,
      account: account,
    })

    return response.accessToken
  } catch (error) {
    console.error('Get Graph token error:', error)
    if (error.name === 'InteractionRequiredAuthError') {
      const response = await msalInstance.acquireTokenPopup(graphScopes)
      return response.accessToken
    }
    throw error
  }
}
```

หลังจากที่ทำการล็อคอินด้วย ms ip จะทำการ redirect ไปตามที่ได้ config เอาไว้

พิงก์ชั่นในการดึงข้อมูลจาก ms graph token

Login with ms ip

azureLogout

```
const logout = async () => {
    localStorage.removeItem('authData')
    if (typeLogin.value == 'AZURE') await azureLogout()
    accessToken.value = ''
    refreshToken.value = ''
    typeLogin.value = ''
    // isLogin.value = false
}
```

```
const azureLogout = async () => {
    if (!msalInstance) {
        throw new Error('MSAL is not initialized')
    }

    await msalInstance.logoutRedirect()
}
```

ฟังก์ชันใช้สำหรับการ Logout ตัว ms

msalConfig

```
const baseUrl = window.location.origin

export const msalConfig = {
    auth: {
        clientId: import.meta.env.VITE_CLIENT_ID,
        authority: 'https://login.microsoftonline.com/' + import.meta.env.VITE_TENANT_ID,
        redirectUri: `${baseUrl}${import.meta.env.VITE_BASE ?? ''}/`,
        postLogoutRedirectUri: `${baseUrl}${import.meta.env.VITE_BASE ?? ''}/`,
        navigateToLoginRequestUrl: true,
    },
    cache: [
        {
            cacheLocation: 'localStorage',
            storeAuthStateInCookie: true,
        },
    ],
}

export const loginRequest = {
    scopes: ['openid', 'profile', 'email', 'User.Read', 'User.ReadBasic.All'],
}

export const graphScopes = {
    scopes: ['User.Read', 'User.ReadBasic.All'],
}

export const graphConfig = {
    graphMeEndpoint: 'https://graph.microsoft.com/v1.0/me',
}

export const state = reactive({
    isAuthenticated: false,
    user: null,
})

export const msalInstance = new PublicClientApplication(msalConfig)
```

ตัว config ต่างๆที่ใช้สำหรับ ms ฟังก์ชันต่างๆ
อย่างเช่นการ redirect

AUTHORIZATION / NAVIGATION GUARD

```
router.beforeEach(async (to, from, next) => {
  const boardApi = useBoardApi()
  const boardStore = useBoardStore()
  const authStore = useAuthStore()
  const taskStore = useTasksStore()
  const statusStore = useStatusesStore()
  let board
  let bStatus

  await authStore.loadAzureData()

  const reset = () => {
    boardStore.resetBoards()
    statusStore.resetStatuses()
    taskStore.resetTasks()
  }

  if (to.meta.requireViewer && to.params.bid) {
    const { response, status } = await boardApi.getBoardById(to.params.bid)
    board = response
    bStatus = status
  }

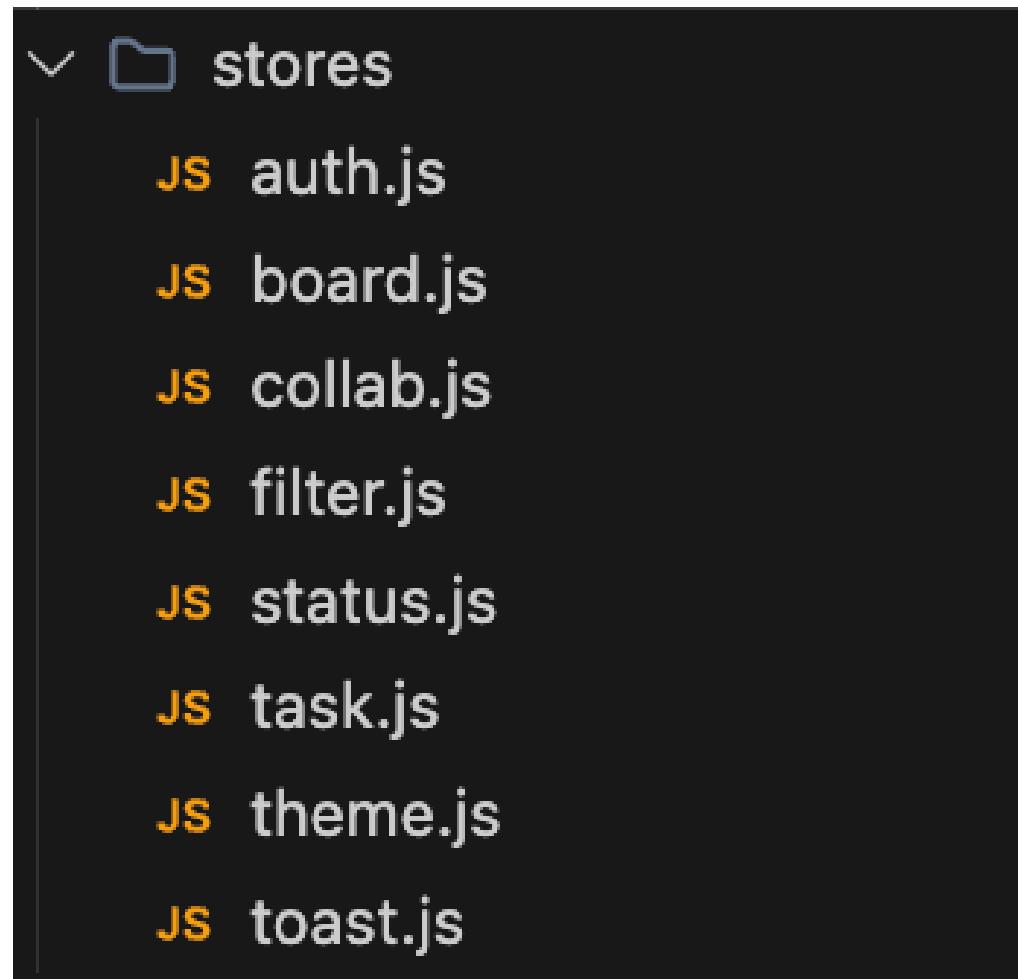
  const isLoggedIn = await authStore.getLoginStatus()
  if (to.meta.requireOwner) {
    if (await authStore.isOwner(to.params.bid)) {
      next()
    } else {
      next({ name: 'accessDenied' })
    }
  } else if (to.meta.requireEditor || (to.meta.requireViewer && bStatus == 403)) {
    if ((await authStore.isOwner(to.params.bid)) || (await authStore.isEditor(to.params.bid))) next()
    else next({ name: 'accessDenied' })
  } else if (!isLoggedIn) {
    if (to.meta.requireLogin || to.meta.requireOwner || to.meta.requireEditor || to.meta.requireViewer) {
      const redirectTo = to fullPath
      next({ name: 'login', query: { redirectTo } })
    } else {
      next()
    }
  }
})
```

```
} else if (isLoggedIn) {
  if (to.name === 'login' && !(await authStore.checkToken())) {
    next({ name: 'boardView' })
  } else if ((from.name === 'login' || from.name === 'boardAdd') && to.name === 'boardView') {
    const result = await boardApi.getAllBoard()
    if (result?.person_boards?.length == 1 && result?.collab_boards == 0) {
      next({ name: 'taskView', params: { bid: result?.person_boards[0].id } })
    } else {
      next()
    }
  } else if (await authStore.checkToken()) {
    if (to.name !== 'login') {
      reset()
      next({ name: 'login' })
    } else {
      next()
    }
  } else {
    next()
  }
} else if (bStatus == 404) {
  reset()
  next({ name: 'notFound' })
} else if (bStatus === 403) {
  reset()
  next({ name: 'accessDenied' })
} else {
  next()
})
```

navigation guard เป็นส่วนหนึ่งของการทำ authorization คือการที่จำกัดสิทธิ์กับ user ต่างๆที่เข้ามาใช้งาน webapp ให้มีสิทธิ์เท่าที่ควรจะได้ มีการเขียน check เอาไว้ต่างๆว่า มีการ login หรือไม่ board เป็นสถาบันรูปแบบไหนหรือ user ที่ล็อกอินเข้ามา มีสิทธิ์เป็น collaborate ใน board นั้นๆหรือป่าว

STATE MANAGEMENT

All stores



Pinia หรือ Store เป็นตัวที่ใช้ในการจัดการ state hook ต่างๆ ซึ่งช่วยในการที่เราสามารถดึงค่าต่างๆ หรือเก็บค่าเอาไว้จัดการผ่านตัว Pinia ได้ ภายในโปรเจ็ค และพิงก์ชั้นที่อาจจะมีการใช้ช้าอยู่บ่อยๆ ไว้ภายใน Pinia และดึงมาใช้เมื่อต้องการใช้ ซึ่ง Pinia ของพวกเรายังคงมีดังนี้

1. auth.js ใช้สำหรับการเก็บข้อมูลของ user
2. board.js ใช้สำหรับการเก็บข้อมูลของ board
3. collab.js ใช้สำหรับการเก็บข้อมูล collaborator
4. filter.js ใช้ในการที่เมื่อ task มีการ filter
5. status.js ใช้สำหรับการเก็บข้อมูล status
6. task.js ใช้สำหรับการเก็บข้อมูล task
7. theme.js ใช้สำหรับการเก็บค่า theme ของตัว webapp
8. toast.js ใช้สำหรับการเก็บพิงก์ชั้น ของ toast เมื่อแอปมีการแจ้งเตือนสิ่งต่างๆ

ตัวอย่างการใช้งาน

```
const boardStore = useBoardStore()
const authStore = useAuthStore()

onMounted(async () => {
    if (authStore.checkToken) await boardStore.fetchBoard()
    if (authStore.checkToken) await collabStore.fetchCollabBoards()
```

env variable , exception handling , components

.env

```
VITE_BASE_URL=http://localhost:8080  
VITE_CLIENT_ID=0c14d0aa-b581-4796-93ed-707ce6c0865e  
VITE_TENANT_ID=79845616-9df0-43e0-8842-e300feb2642a  
|
```

env สำหรับการเก็บ env เวลาที่ test

.env.production

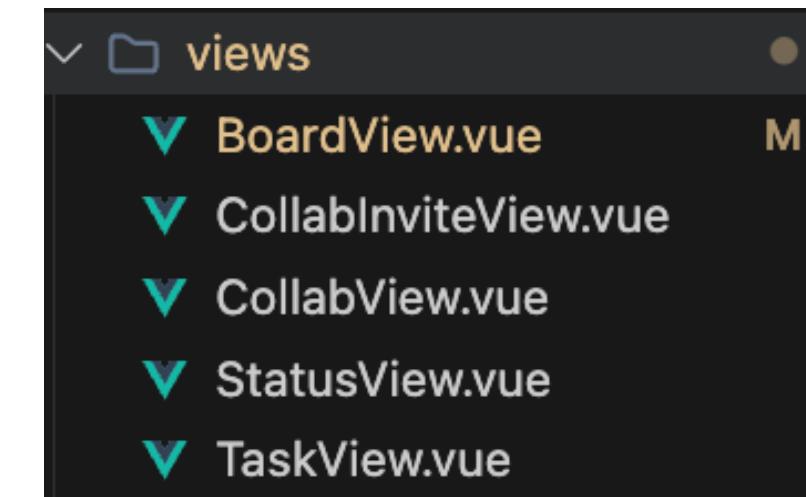
```
VITE_BASE_URL=/sj1/api  
VITE_BASE=/sj1  
VITE_CLIENT_ID=0c14d0aa-b581-4796-93ed-707ce6c0865e  
VITE_TENANT_ID=79845616-9df0-43e0-8842-e300feb2642a
```

env สำหรับการเก็บ env เวลาที่ deploy

components

- ▼ AccessDenied.vue
- ▼ BoardModal.vue
- ▼ BoardTable.vue
- ▼ CollabBoardTable.vue
- ▼ CollaboratorModal.vue
- ▼ ConfirmDelete.vue
- ▼ FilterModal.vue
- ▼ HomePage.vue
- ▼ NavHeader.vue
- ▼ PageNotFound.vue
- ▼ SignIn.vue
- ▼ StatusModal.vue
- ▼ StatusSetting.vue
- ▼ StatusTable.vue
- ▼ TaskModal.vue
- ▼ TaskTable.vue
- ▼ VisibleModal.vue

views



```
views  
  ▼ BoardView.vue  
  ▼ CollabInviteView.vue  
  ▼ CollabView.vue  
  ▼ StatusView.vue  
  ▼ TaskView.vue
```

views คือหน้าที่เรียกใช้ component ต่างๆ

การแปง component บุ่งเนื้องเรื่องของการ reuseable เพื่อให้จำวน component นั้น น้อยลง

env variable , exception handling , components

toast

```
export const useToast = defineStore('toast', () => {
  const route = useRoute()
  const currToast = ref({ style: '', header: '', msg: '' })
  const currToastArr = ref([style: '', header: '', msgArr: []])
  let myTimeout = null
  const changeToast = async (status, head, msg) => {
    resetToast()
    clearTimeout(myTimeout)
    currToast.value.header = head
    currToast.value.style = status == 'success' ? 'alert-success' : status == 'error' ? 'alert-error' : status == 'load' || 'warn' ? 'alert-warn' : 'alert-info'
    if (!status || msg.toLowerCase().includes('transferred') || msg.toLowerCase().includes('the status has been updated.')) {
      if (msg.toLowerCase().includes('task') || msg.toLowerCase().includes('the status does not exist.') || msg.toLowerCase().includes('not found')) {
        await useTasksStore().fetchTasks(route.params.bid)
      }
      if (msg.toLowerCase().includes('status') && !msg.toLowerCase().includes('the status has been updated.')) {
        await useStatusesStore().fetchStatuses(route.params.bid)
      }
    }
    currToast.value.msg = msg
    myTimeout = setTimeout(() => {
      resetToast()
    }, 5000)
  }
})
```

ใช้ toast ในการแจ้งเตือน user ว่าเกิดอะไรขึ้นกับการกระทำ
นั้นๆ เช่น การที่มี toast สีแดงแล้วแจ้งเตือนว่าการสร้าง
task
ไม่สำเร็จ

router

```
  TOR Command, #L TOR Cascade
  [
    {
      path: '/access-denied',
      name: 'accessDenied',
      component: AccessDenied,
    },
    {
      path: '/not-found',
      name: 'notFound',
      component: PageNotFound,
    },
  ]
```

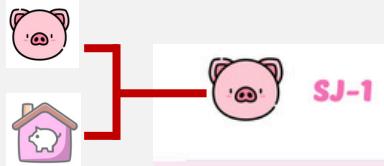
การ handle ไว้ที่ router เมื่อพยายามจะเข้าถึงที่
ไม่มีสิทธิหรือว่าหน้าที่ไม่พบ

UX/UI

COLOR

Logo in homepage

For back to Homepage



The Meaning of The Color Pink

Pink color represents **love, cheerfulness, and creativity**. It is a color that brings calm and can be used in your UI design to create a playful atmosphere.

The color of the toast will change depending on the type that the system wants to notify the user.

● Success

● Error

● Warning

SIGN IN

@ Username

🔒 Password

LOGIN

MICROSOFT LOGIN

Welcome To ITB-KK

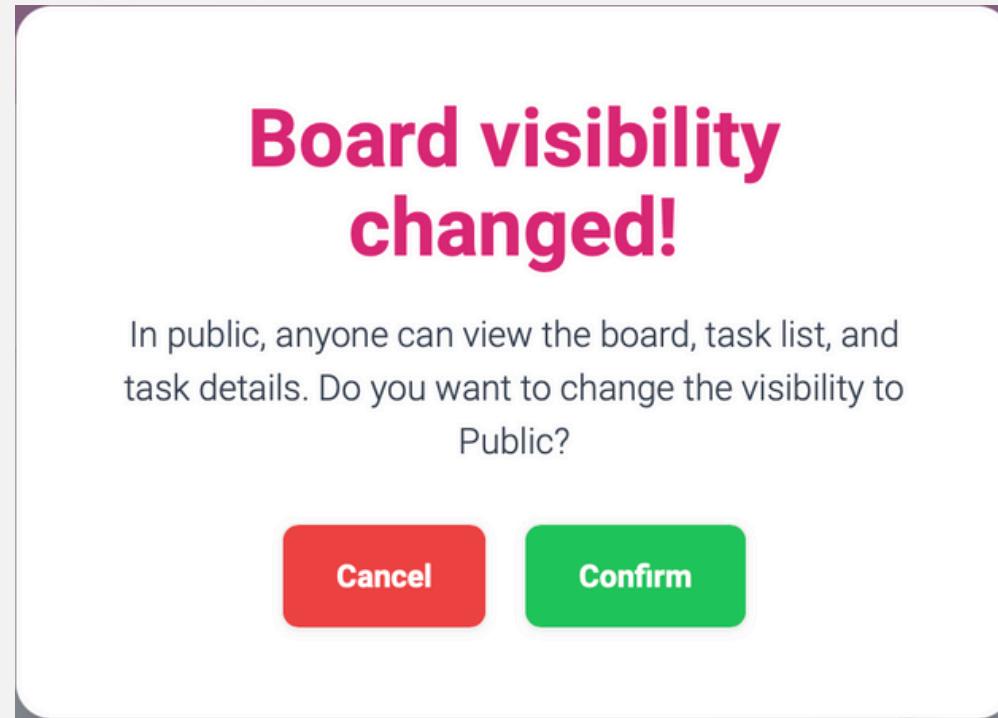
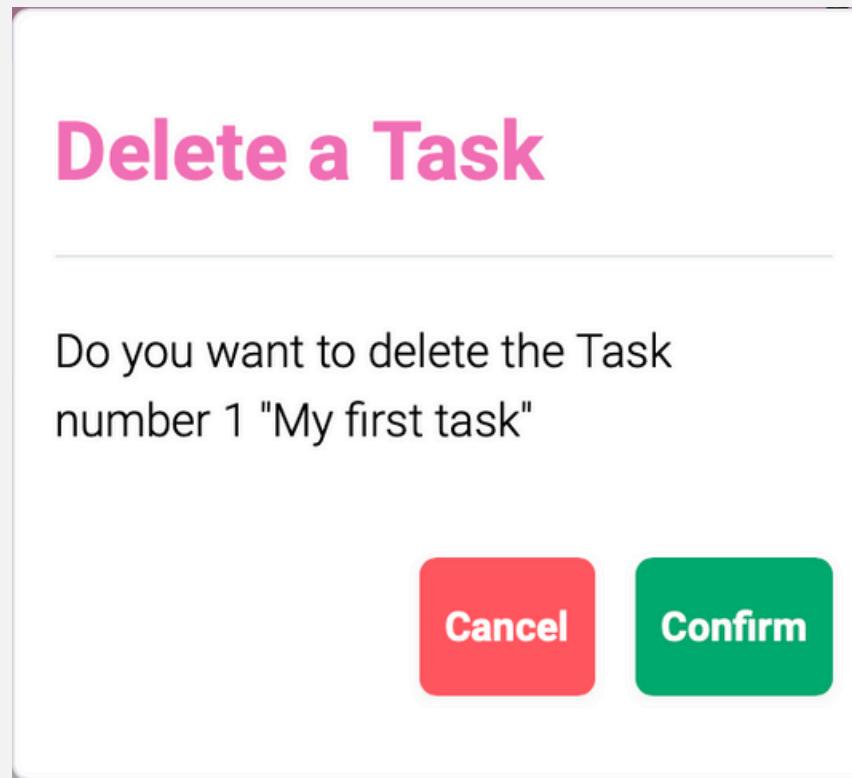
This website is a Kanban board platform designed to empower you to easily visualize your tasks, manage work-in-progress, and enhance workflow efficiency.

Warning: Each task can have at most 10 files. The following files are not added: IP23-requirements.xlsx, IT poster image.png, Ruby on Rail workshop.jpg, SIT history.pdf, SIT history.rtf, SIT logo description.txt, SIT LOGO.png, SIT_VirtualBG_01.jpg, twentymegabytes, สีประจำคณะเทคโนโลยีสารสนเทศ.docx, File with the same filename cannot be added or updated to the attachments. Please delete the attachment and add again to update the file.,

Success: The task has been deleted X

Error: Username or Password is incorrect X

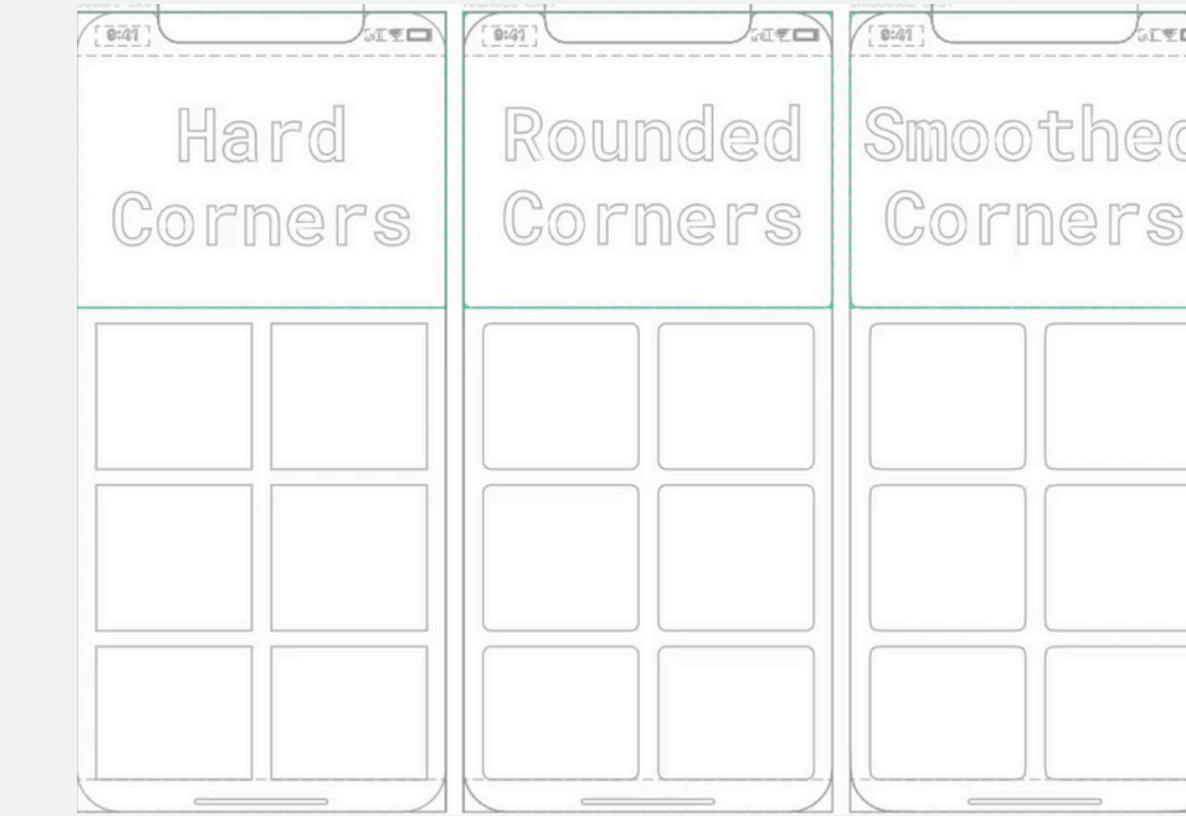
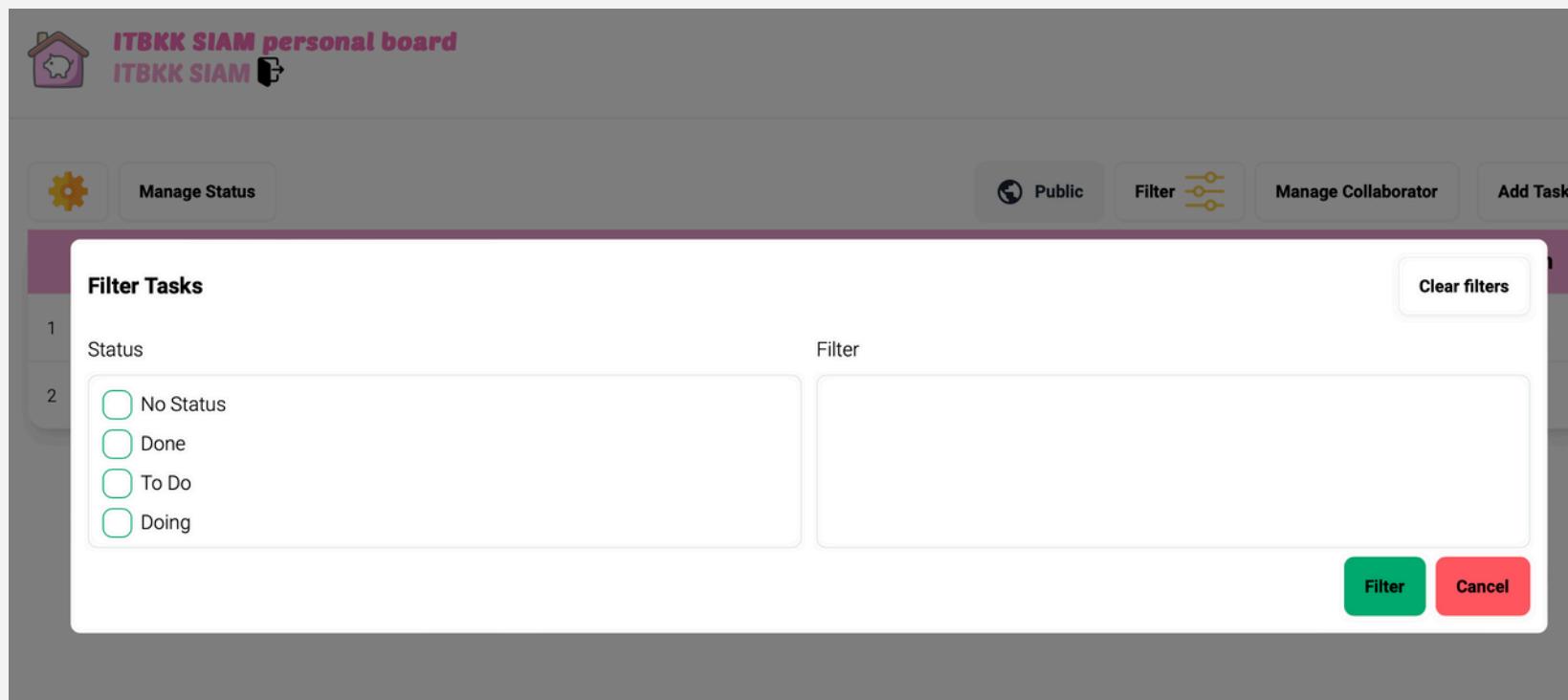
BUTTON

A screenshot of a web-based task management application. The form is titled "Edit Task" at the top left. It includes fields for "Title*" (containing "My first task"), "Description" (with placeholder text "Write your description"), and "Attachments" (with a "Choose Files" button and a note "no files selected"). On the right side, there are sections for "Assignees" (with a placeholder "Enter assignees") and "Status" (with a dropdown menu showing "No Status"). At the bottom right, there are two buttons: a gray "save" button and a red "cancel" button. In the top right corner of the form area, there is a small blue button labeled "Limit disable".

Choose button colors based on the call to action and the emotion we want to motivate user.

- **Grey:** disabled;
- **Green:** positive;
- **Red:** negative;

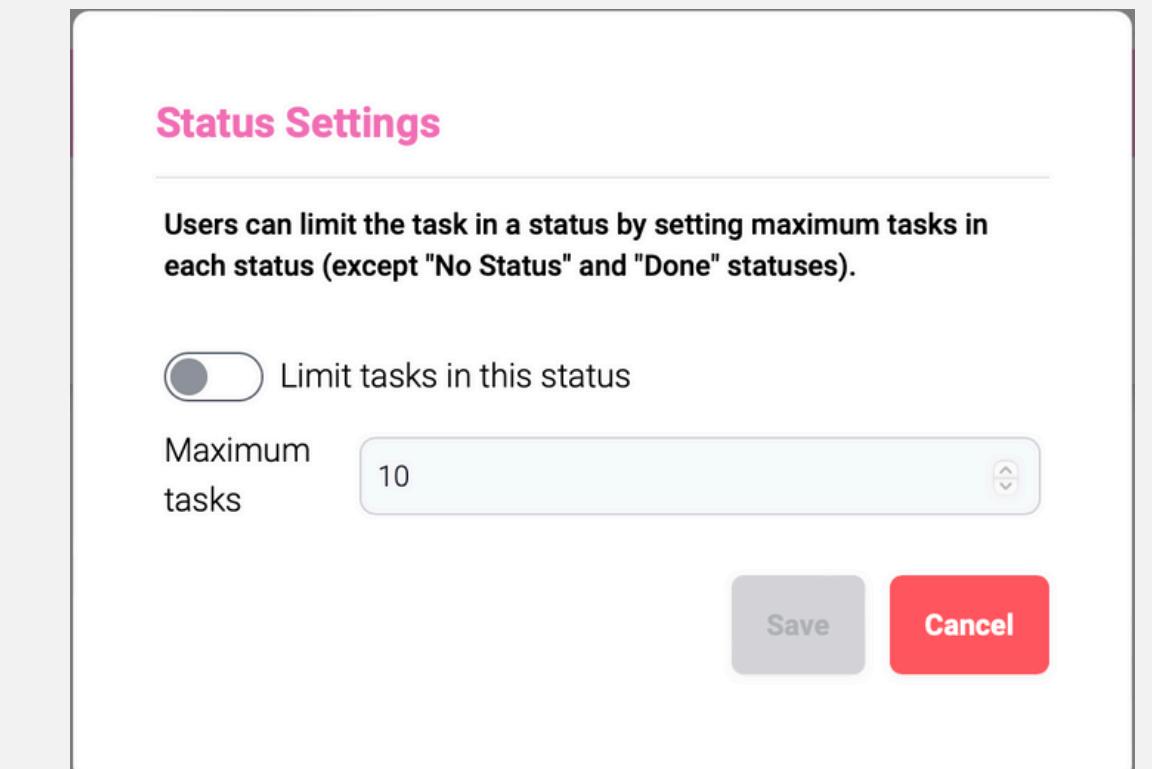
ELEMENT SHAPE



the corner smoothing because it allows the user eye to transition more easily from element to element.

TOGGLE SWITCH

The status setting is designed as a toggle so that users can switch between on and off the limit. The current status is clearly displayed and can be switched by clicking or sliding.



LAYOUT

The screenshot shows the ITBKK SIAM personal board interface. At the top left is a house icon and the text "ITBKK SIAM personal board". Below it is "ITBKK SIAM" with a user icon. To the right are "Logout" and a red-bordered trash can icon labeled "Delete board". The main area has tabs for "Manage Status" (with a gear icon) and "Private" (with a lock icon). There are also "Filter" (with a yellow switch icon), "Manage Collaborator" (with a person icon), and "Add Task" buttons.

The central part is a table titled "Title" with columns for "Assignees", "Status", and "Action". It lists two tasks:

Title	Assignees	Status	Action
1 My first task	Unassigned	No Status	
2 My second task	Unassigned	No Status	

A red bracket on the left points to the table rows, and another red bracket on the right points to the edit icons in the "Action" column. A callout box on the far right says "Edit / Delete each task".

Below the table, a text box states: "Each task is arranged in a table format to make it easy for users to understand."

The screenshot shows two parts of the ITBKK SIAM personal board. On the left is "Collaborator Management" with a table:

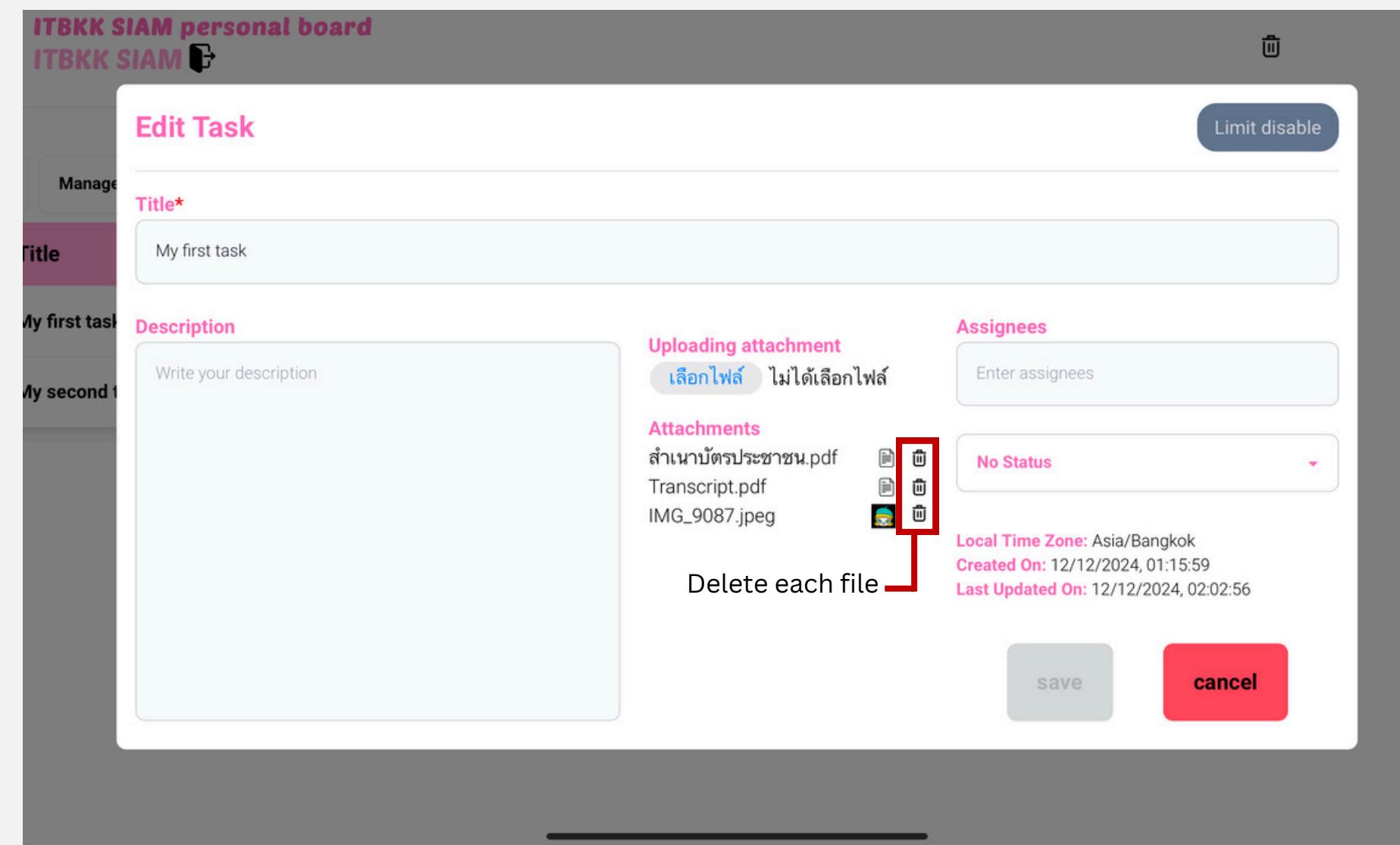
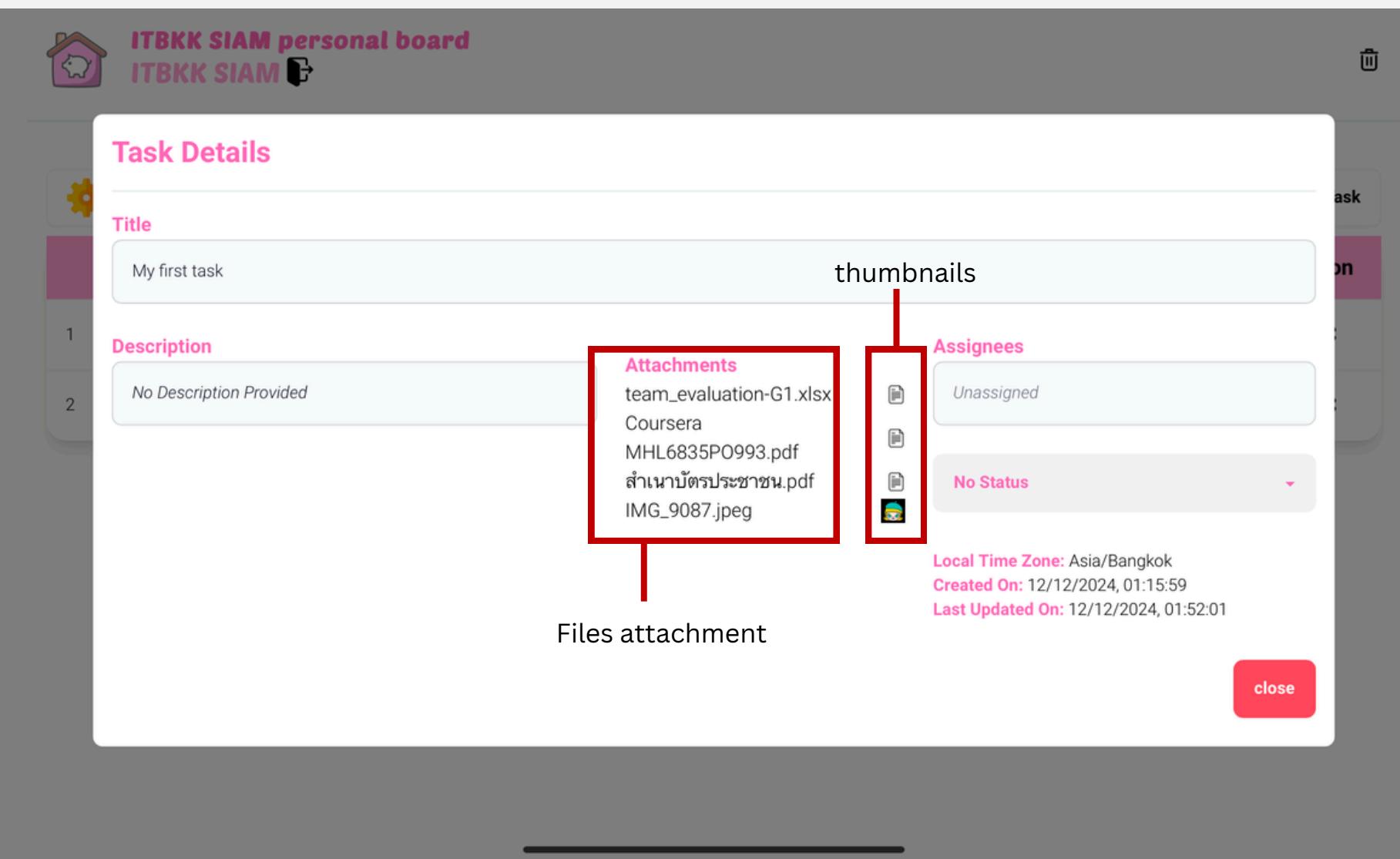
No	Name	Email	Status	Access Right	Action
1	YADA JENRUNGRODSAKUL	65130500017@ad.sit.kmutt.ac.th	PENDING	READ v	
2	NATTHAPHOL NIRATTISAIKUL	65130500105@ad.sit.kmutt.ac.th	PENDING	WRITE v	
3	NIRAPHAN DUANGCHAIRAKKASEM	65130500039@ad.sit.kmutt.ac.th	PENDING	READ v	

On the right is "ITBKK SIAM personal board" with a table titled "Manage Task":

Name	Description	Action
No Status	The default status	
Done	Finished	
To Do	The task is included in the project	
Doing	The task is being worked on	

Each button has a feature clearly written on it so that the user knows what the button does.

LAYOUT Attachment file



LAYOUT

pending invite

invite page

SJ-1
YADA JENRUNGRODSAKUL ↗

Personal boards

Create personal board

No	Name	Status	Action
1	YADA JENRUNGRODSAKUL personal board	PRIVATE	⚙️

Collab Boards

No	Name	Owner	Access Right	Action
1	ITBKK SIAM personal board (Pending Invite)		READ	Accept/Decline

Click for redirect to invite page

YADA JENRUNGRODSAKUL ↗

Thursday, December 12, 2024

Pending invite

S
ITBKK SIAM has invited you to collaborate with READ access right on "ITBKK SIAM personal board"
You can accept or decline this invitation.

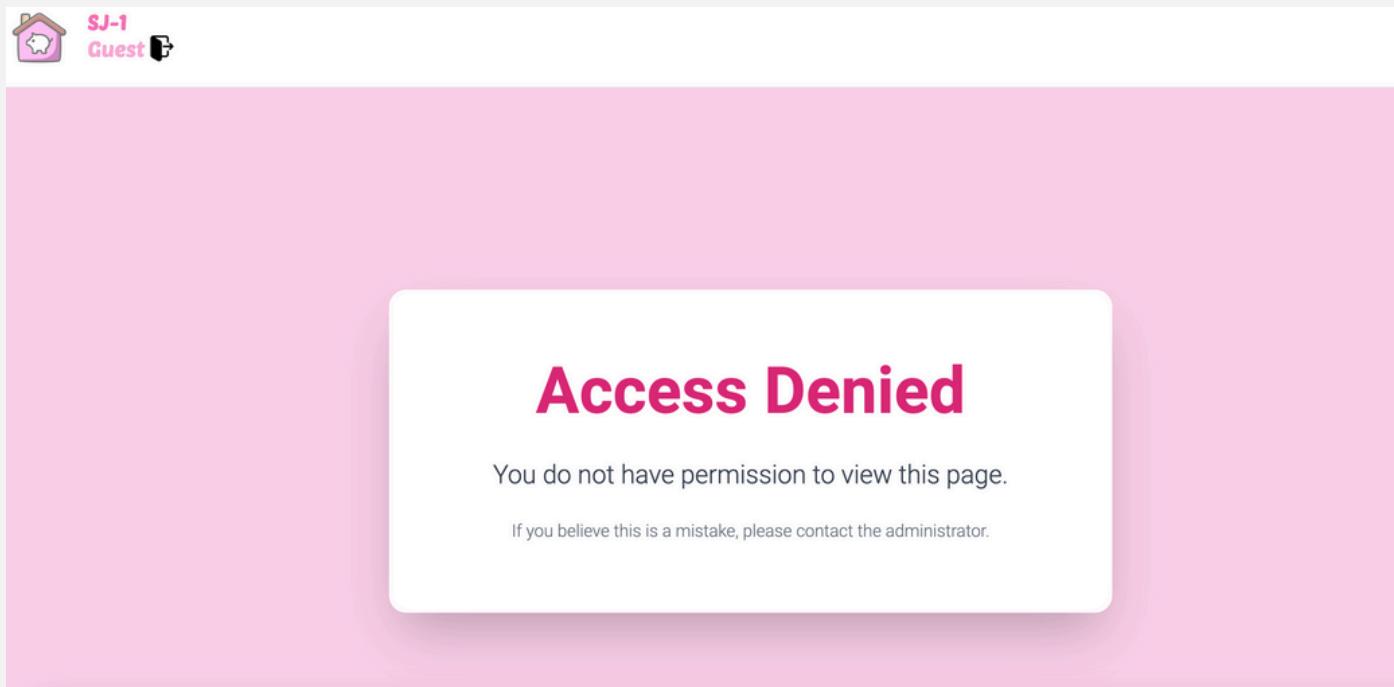
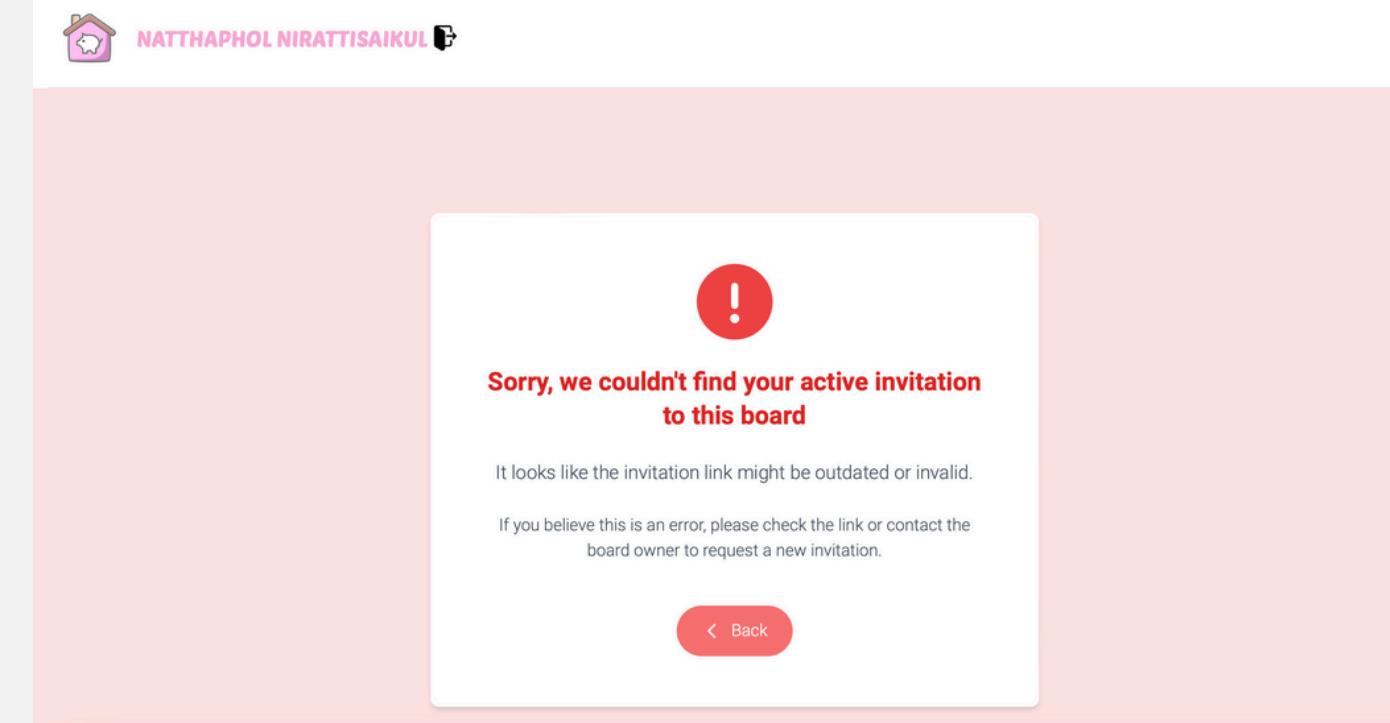
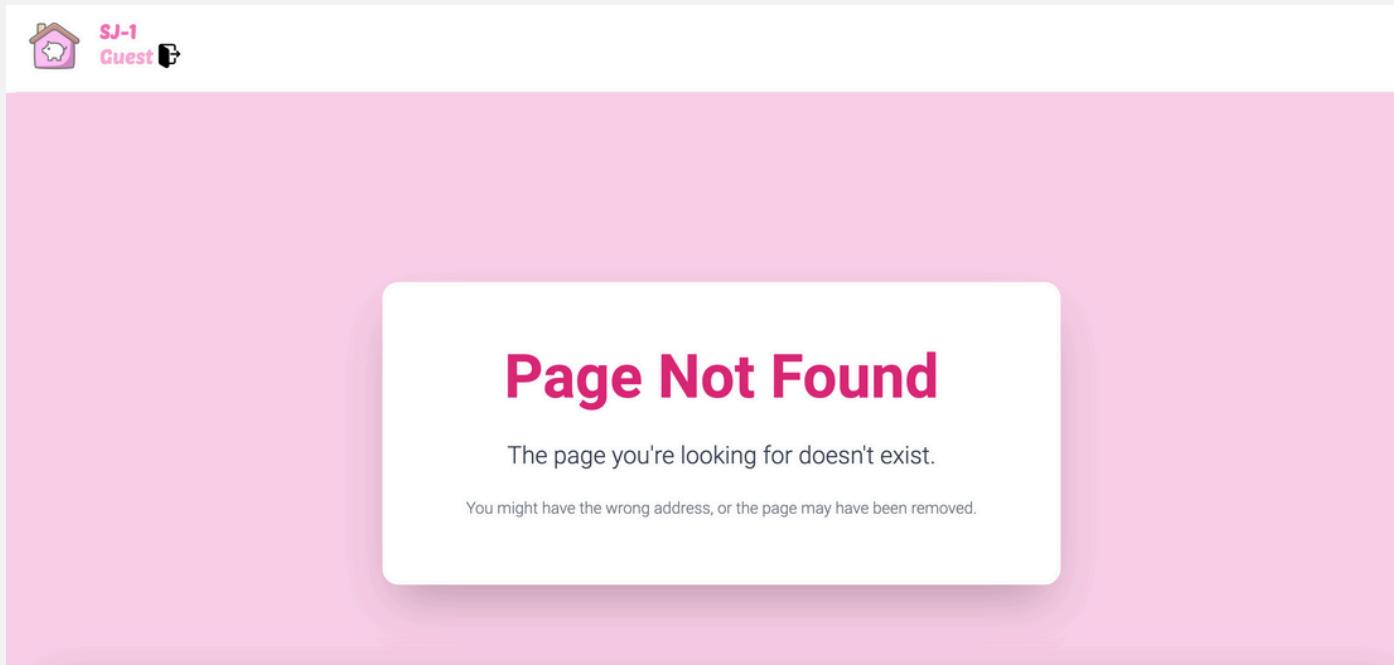
1 members have already become collaborators on this board.

Decline Accept invitation

Show the number of member in that board with random name

LAYOUT

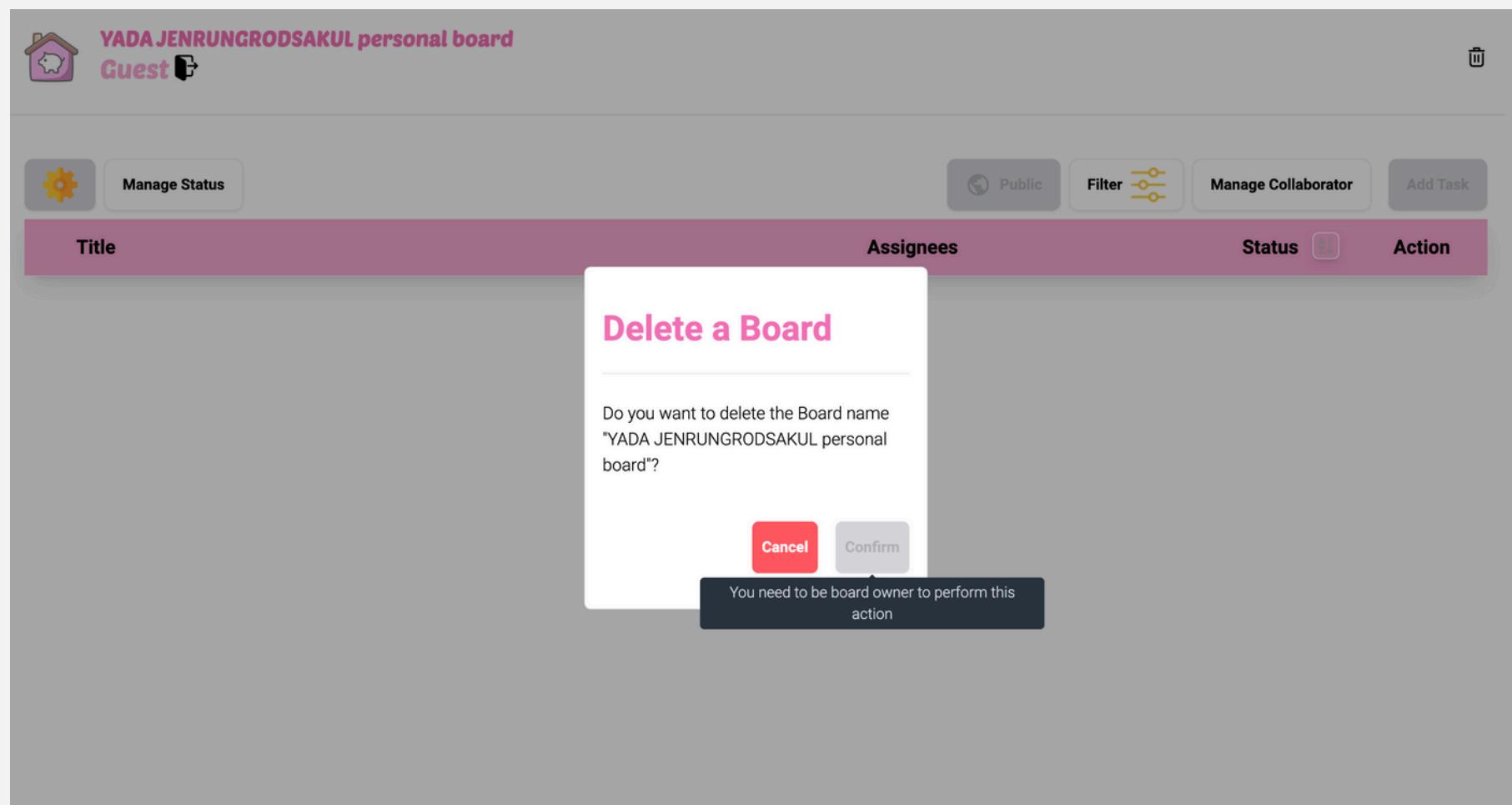
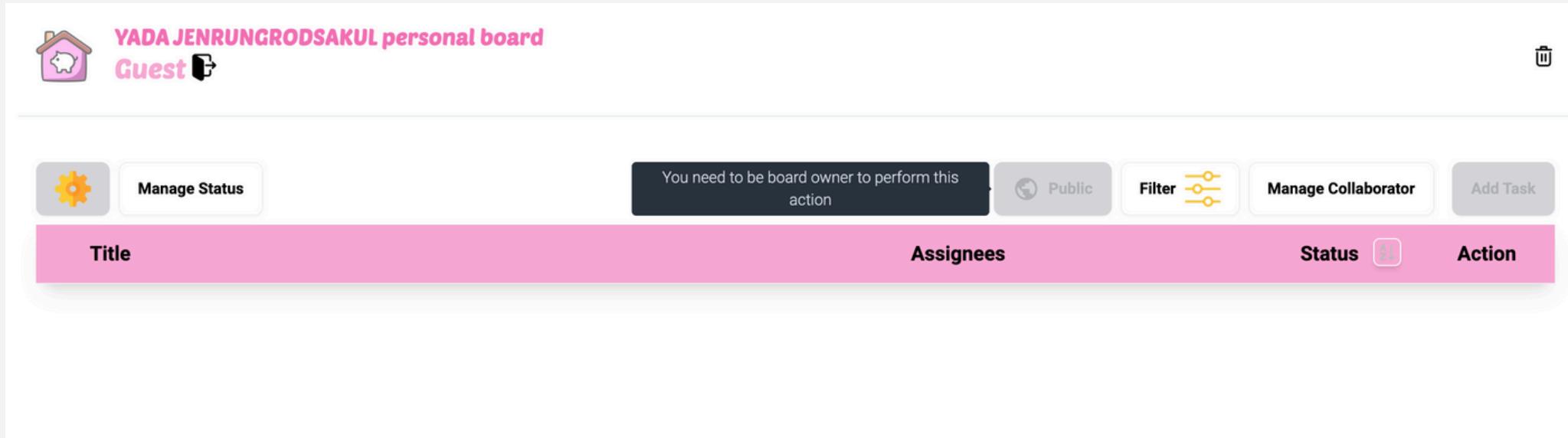
error pages



Use an icon and color that indicates an error page.

LAYOUT

tooltip



There is a tool-tip that tells the user why the button is disabled.