* Add attendence sheet api

Global input-

local storgae - ’Subject\_Names<{[subject: string]: string;}>’

Output-

page=teacher&action=add\_subjects`

Admin\_Sheet\_Id,email,Subject\_Names<string[]>

Response-

(‘sheet\_added’)-’Subject\_names<{‘subject name’:’sheetid’}>’ received from response, add updated subject sheet array in localstorage ’Subject\_Names’

Want-

Add a new sheet ‘subject’ and add sheet details in object in main DB subject column of teacher [{subject sheet name , sheet id and last filled column number}]

* Take attendance api

Global Input-

Local storage- ‘Subject\_Names<{[subject: string]: string;}>’,

’Students\_Dataset<{label:number,descriptor:number[]}[]>’

Output-

page=teacher&action=upload\_attendance

email,Admin\_Sheet\_Id,Roll\_Numbers<String[]>,Subject<{subject:string,id:strig}>, Column<’New’|’Previous’>

Response-

(‘attendance\_added’)-Navigate to /teacher

Want-

get last filled coloum number from DB from subject object,then add a coloum on last then assign attendance align to their roll number on last filled coloum + 1

* Syncing subject names api

Global Input-

Sessionstorage- email,Admin\_Sheet\_Id

Output-

page=teacher&action=sync\_subjects\_name

email,Admin\_Sheet\_Id,available\_subjects<number>

Response-

(‘Subjects’)-add ‘Subjects’ data in local storage received from response

(‘up\_to\_dated’)-return

Want-

If available subjects length 0 or unequal to subject inside DB then give all subjects names of teacher

* Syncing imgs api

Global Input-

Sessionstorage- email,Admin\_Sheet\_Id

Localstorage- Student\_Ids<string[]>

Output-

page=teacher&action=sync\_student\_imgs\_ids

Admin\_Sheet\_Id,available\_student\_ids<string[]|false>

Response-

(‘student\_imgs’)-’student\_imgs’,’student\_ids’ data received from response update in local storage ‘Students\_Dataset’,’Student\_Ids’

Want-

If available\_student\_ids false then send all imgs with id else give ids which not in available

* login api

Output-

page=${selectedRole}&action=login

email,password,username,selectedRole<admin|teacher|student>

Response-

(‘set\_password’)-set session ‘Admin\_Sheet\_Id’

(‘logind’)-

set ‘Admin\_Sheet\_Id’,’token’ received from response ,email ,’sheet\_exist’ ,’user’=teacher to session storage

Want-

Get admin sheet id from username key from my sheet and if sheet exist with access then send response admin sheet id ,if password is null then check prepassword if match then send ‘message’ to set passwrod

* Teacher set password api

Global input-

Session storage-Admin\_Sheet\_Id

Output-

page=${selectedRole}&action=set\_password

Newpassword,prepassword,Admin\_Sheet\_Id,email,selectedRole

Response-

(‘message’==”password\_set”)-set session email,’sheet\_exist’,user,token

Want-

Check if prepassswod equal then Set new passoword in password and set null to prepassword colum to selected Role in AdminSheet or my DB

* Sheet access check api

Global input-

Session storage-’sheet\_exist’

Output-

page=admin&action=checking\_admin\_sheet\_access\_valid

Admin\_Sheet\_Id

Response-

(‘sheet\_invalid’)

(‘Sheet\_valid’)-set session to “sheet\_exist” navigate to teacher

Want-

Check sheet valid and has access

* Common data from server
  + ‘sheet\_invalid’-when sheet invalid then this var in response
  + ‘error’-when server error then receive in response
  + ‘message’-on every response from server
* Signup api

Output-

?page=admin&action=signup

name,email,username,password (size 50)

Response -

(‘account\_created’)-navigate to login

Want-

Create account of admin

* Forget password api

page=${selectedRole}&action=send-forget-password-email

email,username,selectedRole

Response-

(‘Sended\_otp’)-set session ‘changePasswordToken’ receive from response

Want-send otp on email and set token and send token to user for change password

* Change password api

page=${selectedRole}&action=change-password

Otp,newPassword,email,username,changePasswordToken

Response-

(‘changed’)-navigate to login page

Want-check token if correct then change the password