



REGISTRATION AUTOMATION

Efficient event registration automation using JavaScript
and Google Apps Script....!!



@Selfmadechouhan



Registration-Automation /RegistrationAutomation.js

```
/* Crafted by me (Priyanshu Chouhan) for the esteemed Rhythm-2024 university fest registration, I present a sophisticated solution to streamline the submission process....
```

```
Within this code, meticulously crafted, I've defined parameters encapsulated within the "RNTU" constant, utilizing Google Apps Script to orchestrate seamless data management.... Through meticulous extraction and generation processes, this script ensures accuracy and efficiency in handling registration numbers and confirmation messages.... */
```

```
// This script automates the process of generating and updating registration numbers in a Google Form and its associated Google Sheet whenever a form submission occurs.... It's particularly useful for managing registration processes or tracking submissions with unique identifiers....
```

```
const RNTU = {  
    // This is an object containing properties related to a registration number.... It includes a header, prefix, and length....  
  
    HEADER: "Registration_Number",  
  
    PREFIX: " RNTU",  
  
    LENGTH: 5,  
  
}  
  
class App {  
  
    // This is a class definition named App.... It contains methods to handle tasks related to a Google Form and its associated Google Sheet....  
  
    constructor() {  
  
        // Initializes the class by obtaining the active spreadsheet and its linked sheet.... It also retrieves the associated form and its confirmation message.... Additionally, it creates a regular expression based on the registration number prefix and length....  
  
        this.ss = SpreadsheetApp.getActive()  
  
        this.sheet = this.getLinkedSheet()  
  
        if (!this.sheet) {  
  
            throw Error(`There is no linked form in this spreadsheet.`)  
        }  
    }
```



@Selfmadechouhan



Registration-Automation /RegistrationAutomation.js

```
this.form = FormApp.openByUrl(this.sheet.getFormUrl())

this.message = this.form.getConfirmationMessage()

this.uidRegex = new RegExp(`^${RNTU.PREFIX}[\d]{${RNTU.LENGTH}}`, 'gi')

}

createUidByNumber(number) {

    // Generates a unique registration number based on a given number by appending the
    prefix and padding the number to match the specified length....

    return RNTU.PREFIX + (10 ** RNTU.LENGTH + number).toString().slice(-RNTU.LENGTH)

}

getLinkedSheet() {

    // Finds and returns the sheet linked to the active spreadsheet....

    return this.ss.getSheets().find(sheet => sheet.getFormUrl())

}

getUidFromConfirmationMessage() {

    // Extracts the registration number from the confirmation message of the form
    submission....

    const message = this.form.getConfirmationMessage()

    const results = message.match(this.uidRegex)

    if (!results) throw Error(`No RNTU found in the current confirmation message with regex
    ${this.uidRegex}.`)

    return results[0]

}
```



@Selfmadechouhan



```
● ● ● Registration-Automation /RegistrationAutomation.js

this.sheet.getRange(rowStart, uidHeaderIndex + 1).setValue(uid)

}

updateConfirmationMessage(nextUid) {

    // Updates the confirmation message of the form by replacing the current registration
    // number with the next one.....

    const message = this.message.replace(this.uidRegex, nextUid)

    this.form.setConfirmationMessage(message)

}

run(e) {

    // Main function that executes when a form is submitted....!! It retrieves the row
    // where the form was submitted, gets the current registration number, saves it in the
    // spreadsheet, generates the next registration number, and updates the form's confirmation
    // message.....

    const {rowStart} = e.range

    const currentUid = this.getIdFromConfirmationMessage()

    this.saveCurrentUid(currentUid, rowStart)

    const nextUid = this.createNextUid(currentUid)

    this.updateConfirmationMessage(nextUid)

}

}

function _onFormSubmit(e) {

new App().run(e)

}

function _onFormSubmit(e) {

new App().run(e)

}
```



@Selfmadechouhan



```
Registration-Automation /RegistrationAutomation.js

/*
----- Instructions for Implementing this Code to Suit Your Needs -----
01). Create a new google form.....
02). Go to response.....
03). Create new Spreadsheet
04). In the Spreadsheet, go to Extensions > Apps Script > Delete the script already
written there, in "Code.gs"
05). Now paste the code written above....!!
06). Save it
07). Now go to tigger (It is on the left bar of the same page)
08). Now, create a tigger.... On the last option select "On form submission" select them
and save it.... You have to allow
pop-up and allow all the settings....
09). Now in the setting tab of the form, go to presentation section & in the response tab
select edit....
10). Type " Form submitted successfully....!! Your Registration No (Unique ID) is: "
RNTU56789 "
( Update Organization Identifier: Replace occurrences of "RNTU" with your
organization's abbreviated name throughout the codebase.... This ensures consistency and
reflects your organization's branding....)
11). Save it & try filling the google form.... It will work....!!

By following these steps, you can effectively utilize the provided code to meet your
specific requirements, facilitating seamless registration processes and enhancing overall
efficiency.....
*/
```

Rhythm 2024
PASS REGISTRATION
Embrace a melodious journey at RNTU Rhythm, where rhythm converges to create unforgettable moments...
Venue: Rabindranath Tagore University Campus

Rhythm RNTU 2024 Registration

Form submitted successfully....!! Your Registration No (Unique ID) is: " RNTU56797 "

This content is neither created nor endorsed by Google. [Report Abuse](#) · [Terms of Service](#) · [Privacy Policy](#)

Google Forms

Update Organization Identifier: Replace occurrences of "RNTU" with your organization's abbreviated name throughout the codebase.... This ensures consistency and reflects your organization's branding....



@Selfmadechouhan



The Google Form Responses Spreadsheet serves as a comprehensive repository for all collected responses, meticulously organized for efficient data management and analysis.... Each response is meticulously arranged in a structured format, facilitating seamless access and interpretation of vital information.... This professional-grade spreadsheet ensures precision and clarity in capturing data from respondents, empowering users to derive valuable insights and make informed decisions based on the collected data....



	A	B	C	D
1	Timestamp	Student Name	College Name	Registration_Number
2	04/04/2024 18:28:03	Priyanshu Chouhan	RNTU	RNTU56789
3	04/04/2024 18:28:47	Abhijeet Suresh	RNTU	RNTU56790
4	04/04/2024 18:30:26	Jitendra Mishra	RNTU	RNTU56791
5	04/04/2024 18:31:56	Madhav Makwana	LNCT	RNTU56792
6	04/04/2024 18:33:13	Prachi Shevgaonkar	RNTU	RNTU56793
7	04/04/2024 18:39:49	James	Princeton University	RNTU56794
8	04/04/2024 18:42:34	Theodore	Yale University	RNTU56795
9	04/04/2024 18:45:17	Akshat Sharma	LNCT	RNTU56796
10	04/04/2024 18:46:43	Lalit Mathankar	LNCT	RNTU56797



@Selfmadechouhan