

Project 250

250 වන ව්‍යාපෘතිය



Coding School



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <link rel="stylesheet"
8     href="imageOpacity.css">
9   <title>Document</title>
10 </head>
11 <body>
12   <div class="demo-wrap">
13     
18     <div class="text">
19       <h1>Hello World!</h1>
20     </div>
```

html

Glitch



වෙබ් පිටුව බලන්න



Glitch ආරම්භ කරන්න

HTML, CSS සහ JavaScript භාවිතයෙන් පහත පරිදි Registration form එකක් සකස් කරන ආකාරය බලමු.

- ❖ 249 ව්‍යාපෘතියේදී නිර්මාණය කරගත් forms වලට database connect කර user registration හා login වලට අදාළ කොටස මෙම ව්‍යාපෘතියේදී ඉගෙනගමු.
- ❖ ප්‍රථමව නිර්මාණය කරගත් forms වල elements සඳහා පහත පරිදි ids add කරගන්න.
- ❖ Sign up form එකෙහි email input tag එක සඳහා පහත පරිදි id එක add කරගන්න.

```
<input id="email" type="text" placeholder="Enter Email" name="email" required />
```

- ❖ Sign up form එකෙහි password input tag එක සඳහා පහත පරිදි id එක add කරගන්න.

```
<input id="password" type="password" placeholder="Enter Password" name="psw" required />
```

- ❖ Sign up form එකෙහි repeat password input tag එක සඳහා පහත පරිදි id එක add කරගන්න.

```
<input id="repeatPw" type="password" placeholder="Repeat Password" name="psw-repeat" required />
```

- ❖ Sign up form එකෙහි cancel button එක සඳහා පහත පරිදි id එක හා onclick event add කරගන්න.

```
<button type="button" class="cancelbtn" id="cancelbtn" onclick="reset()">  
Cancel  
</button>
```

- ❖ Sign up form එකෙහි sign up button එක සඳහා පහත පරිදි id එක හා onclick event add කරගන්න.

```
<button type="button" class="signupbtn" id="signupbtn" onclick="register()">  
Sign Up  
</button>
```

- ❖ Login form එකෙහි username input tag එක සඳහා පහත පරිදි id එක add කරගන්න.

```
<input id="username" type="text" placeholder="Enter Username" name="uname" required />
```

- ❖ Login form එකෙහි password input tag එක සඳහා පහත පරිදි id එක add කරගන්න.

```
<input  
type="password"  
placeholder="Enter Password"  
name="psw"  
required  
id="psw"  
>
```

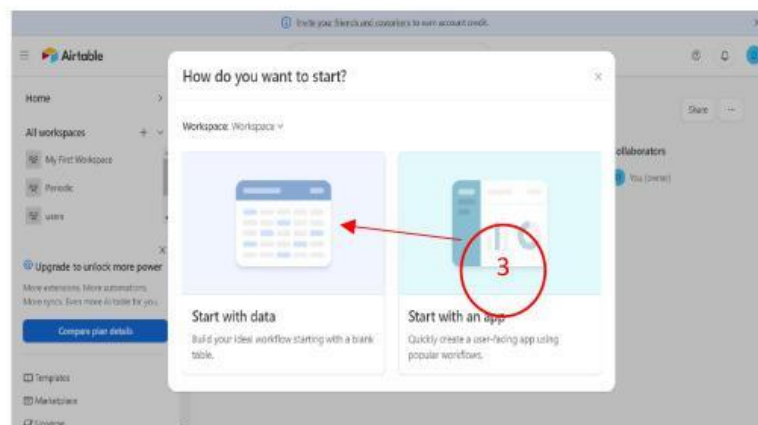
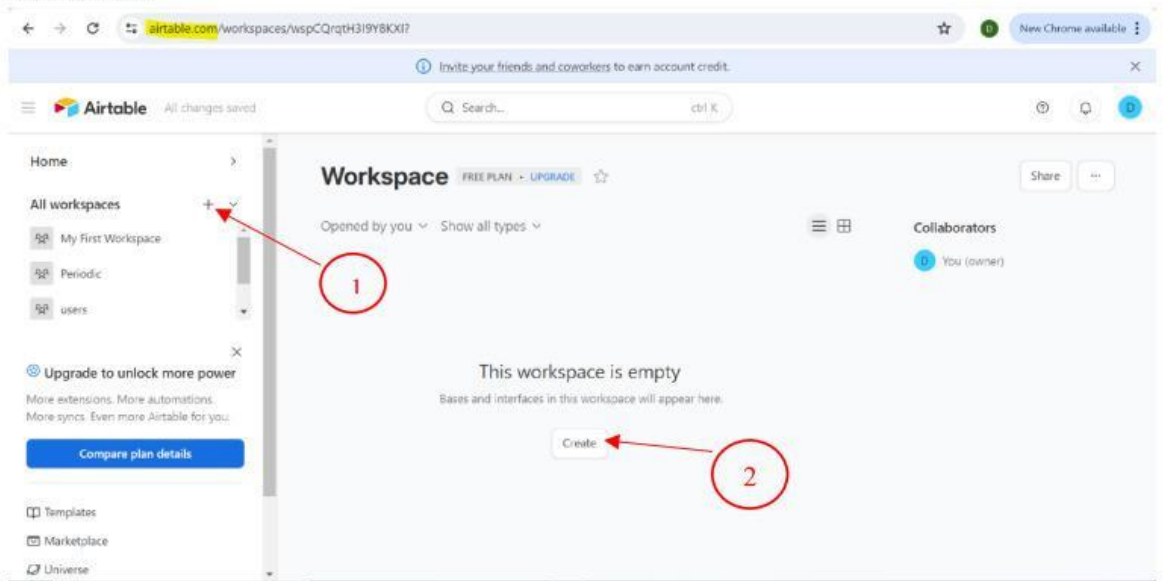
- ❖ Login form එකෙහි login button එක සඳහා පහත පරිදි id එක හා onclick event add කරගන්න.

```
<button id="submitbtn" type="button" onclick="login()">Login</button>
```

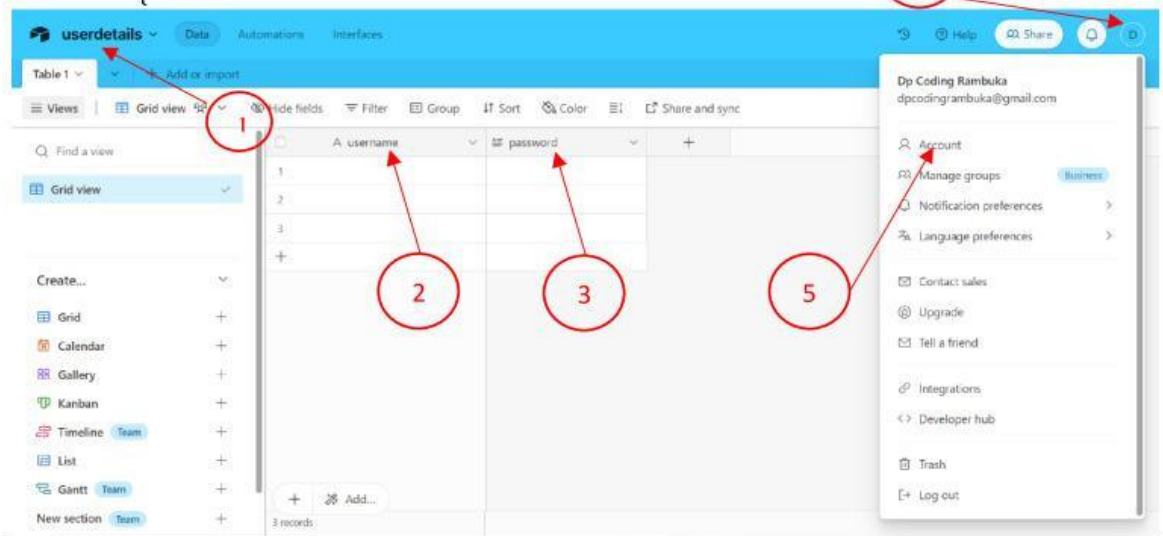
- ❖ Form tag එක පහත පරිදි සකස් කරගන්න. එහි ඇති සියලු attributes clear කරන්න.

```
<form>
```

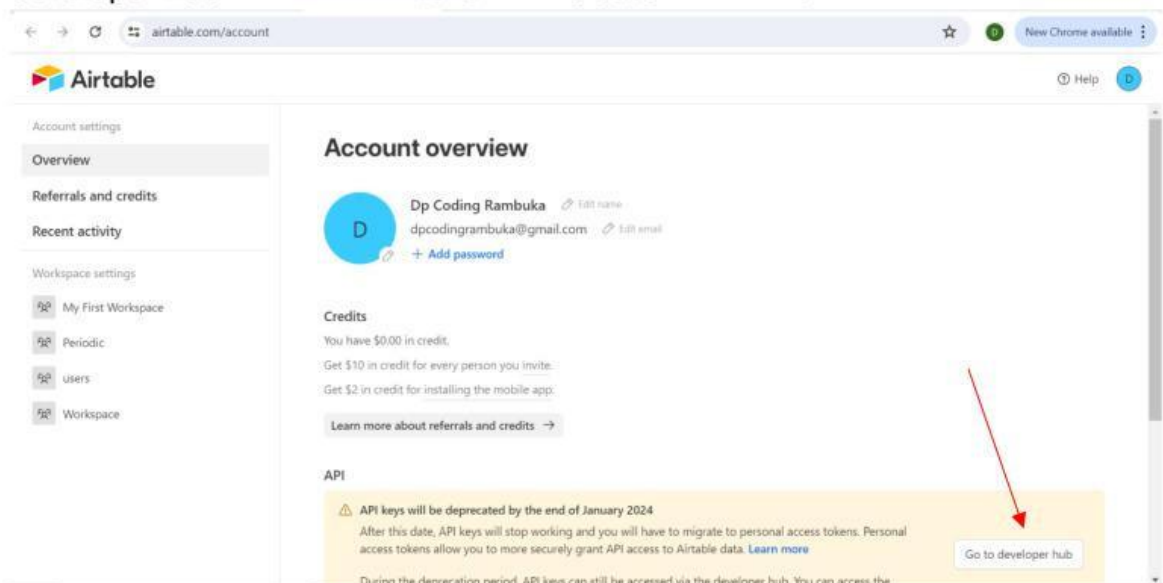
- ❖ ඇත් airtable.com website එකට ගොස් පහත පරිදි new table එකක් create කරගන්න.



❖ පහත පරිදි table එක edit කරගන්න.



❖ Account වෙතට ගිය විට පහත පරිදි open වන web page එක තුළ ඇති Go to developer hub වෙතට ගොස් token එක create කරගන්න.



❖ Create token මත click කරන්න.

API key

Back to home

API keys are deprecated

As of February 1st 2024, Airtable API keys are deprecated and no longer work. Instead of using an API key, you need to migrate to personal access tokens, which allow you to more securely grant API access to Airtable data. [Learn more](#)

Create token

❖ Create new token මගින් අලුත් token එකක් create කරන්න.

Search...					Create new token
NAME	ACCESS	SCOPES	TOKEN ID	DATE CREATED	
periodic	None	data.records:read	patF8qVUy1C0B091	May 8, 2024	
periodic_table	All workspaces and bases	data.records:read	patC7oX18pTy8Uaz	May 9, 2024	
users	1 workspace	data.records:read and 3 more	patC81oR50Cj4ePrj	May 28, 2024	

❖ Token එකට name, scope හා base එක පහත ආකාරයට සකස් කරන්න.

Name

The token's name will be visible in record revision history.

userdetails

පහත සියලුම scopes select කරන්න.



නිර්මාණය කරගත් Workspace එකට අදාළව ඊට ඉහතින් ඇති access එක select කරන්න.

WORKSPACE

👤 All current and future bases in this workspace

Us userdetails

Name

The token's name will be visible in record revision history.

userdetails

Scopes ⓘ

With this token, you will be able to:

data.records:read

See the data in records

data.records:write

Create, edit, and delete records

data.recordComments:read

See comments in records

data.recordComments:write

Create, edit, and delete record comments

+ [Add a scope](#)

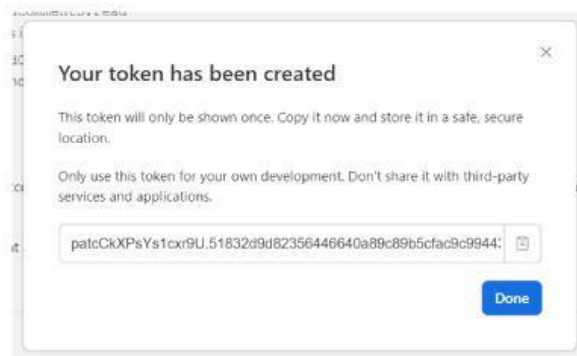
Access ⓘ

This token can access the following bases and workspaces. You can only grant access to bases and workspaces you have access to

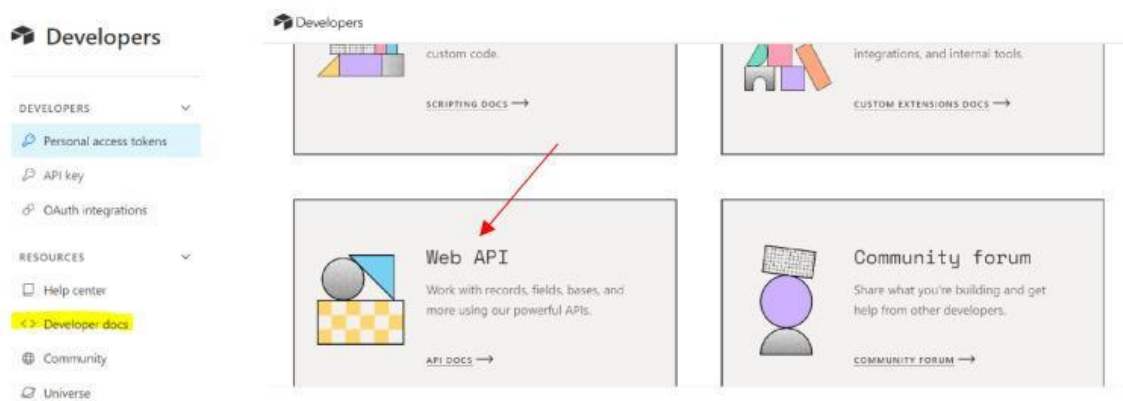
WORKSPACE

👤 All current and future bases in this workspace

- ❖ පහත ආකාරයට generate වන token එක copy කරගත එය notepad එකක save කරගන්න. මෙම token එක පැමිණෙන්නේ එක් වරක් පමණි.



- ❖ Developer docs වෙත යන්න. ඉන්පසු web page එකෙහි webApi වෙත යන්න.

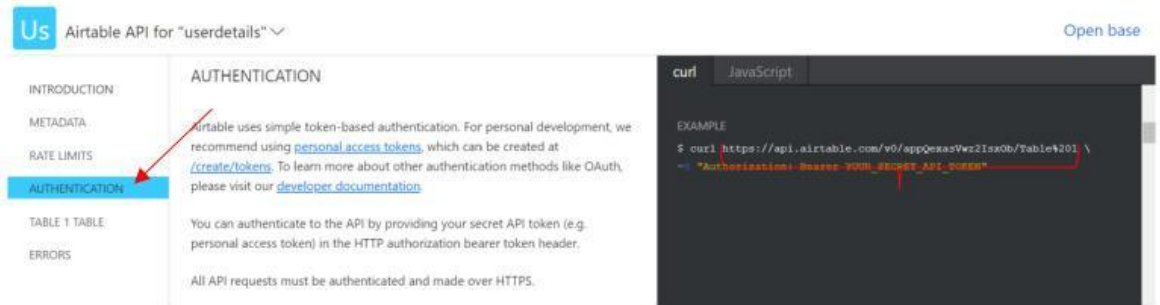


- ❖ පහළට scroll කර නිර්මාණය කරගත් data table එක සොයා එය මත click කර එහි documentation එක open කරගන්න.

To view API documentation that is generated for a particular base, see below:



- ❖ Authentication වෙත ගොස් apiurl එක copy කරගන්න.



- ❖ දැන් script.js file එකට ගොස් පහත පරිදි variable දෙකක ඉහතදී copy කරගත් api token එක හා api url එක ඒවාට add කරගන්න.

```
const apiKey =  
  "patCSICR52CyfzTrj.ee2913ca4c60be08f8e31523dbff018e0e7d0fb82189edb25027d227a4cdf86b";  
const baseUrl = "https://api.airtable.com/v0/appyHtUzXQg4zzp1g/Table%201";
```

- ❖ Sign up button එක click කළ විට call වන register function එක පහත පරිදි සකස් කරගන්න.

```
function register() {  
  const email = document.getElementById("email").value;  
  const password = document.getElementById("password").value;  
  const repeatPw = document.getElementById("repeatPw").value;  
  
  if (email === "" || password === "" || repeatPw === "") {  
    alert("All fields are required!");  
  }  
  if (password !== repeatPw) {  
    alert("Passwords do not match!");  
  }  
  var newUser = {  
    fields: {  
      email: email,  
      password: password,  
    },  
  };  
  const headers = {  
    Authorization: `Bearer ${apiKey}`,  
    "Content-Type": "application/json",  
  };  
  fetch(baseUrl, {  
    method: "POST",  
    headers: headers,  
    body: JSON.stringify(newUser),  
  })  
    .then((response) => response.json())  
    .then((data) => {  
      console.log("Successfull");  
      window.open("login.html");  
    });  
}
```

- ❖ Login button එක click කළ විට call වන login function එක පහත පරිදි සකස් කරගන්න.

```
function login() {
  const username = document.getElementById("username").value;
  const psw = document.getElementById("psw").value;
  if (username === "" || psw === "") {
    alert("All fields are required!");
  }
  const headers = {
    Authorization: `Bearer ${apiKey}`,
    "Content-Type": "application/json",
  };
  console.log(username, psw);

  fetch(`${baseUrl}?filterByFormula={email}='${username}'`, {
    method: "GET",
    headers: headers,
  })
    .then((response) => response.json())
    .then((data) => {
      if (data.records.length === 0) {
        alert("User not found!");
        return;
      }

      const userRecord = data.records[0];
      const storedPassword = userRecord.fields.password;

      if (psw === storedPassword) {
        window.open("home.html", "_self");
      } else {
        alert("Incorrect password!");
      }
    })
    .catch((error) => {
      console.error("Error fetching user data:", error);
      alert("An error occurred. Please try again later.");
    });
}
```

- ❖ Cancel button එක click කළ විට call වන reset function එක පහත පරිදි සකස් කරගන්න.

```
function reset() {
  document.getElementById("email").value = "";
  document.getElementById("password").value = "";
  document.getElementById("repeatPw").value = "";
}
```

- ❖ ඉහත පරිදි නිර්මාණය කරගත් පසු database එක තුළ register වන users සියල්ල save වන අතර login form එක මගින් එසේ save වූ user details enter කර log විය හැකිය.