TY B.Tech Computer Engineering

2018 Course

CLOUD COMPUTING SCE

SUBMITTED TO:

PROF. VISHAL MESHRAM

VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY, PUNE

COMPUTER ENGINEERING DEPARTMENT

Anish Mudaliar	**321002**	**21810945**
Amlan Nanda	**321032**	**21810954**
Amey Bobade	**321007**	**21810866**
Pritesh Bhutada	**321042**	**21810832**

So in our SCE we have discussed about 3 Cloud Services:

1. Hosting a website using MICROSOFT AZURE

At its core, **Azure** is a public cloud computing platform—with solutions including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) that can be **used for** services such as analytics, virtual computing, storage, networking, and much more.

2. Login Portal Authentication using GOOGLE FIREBASE

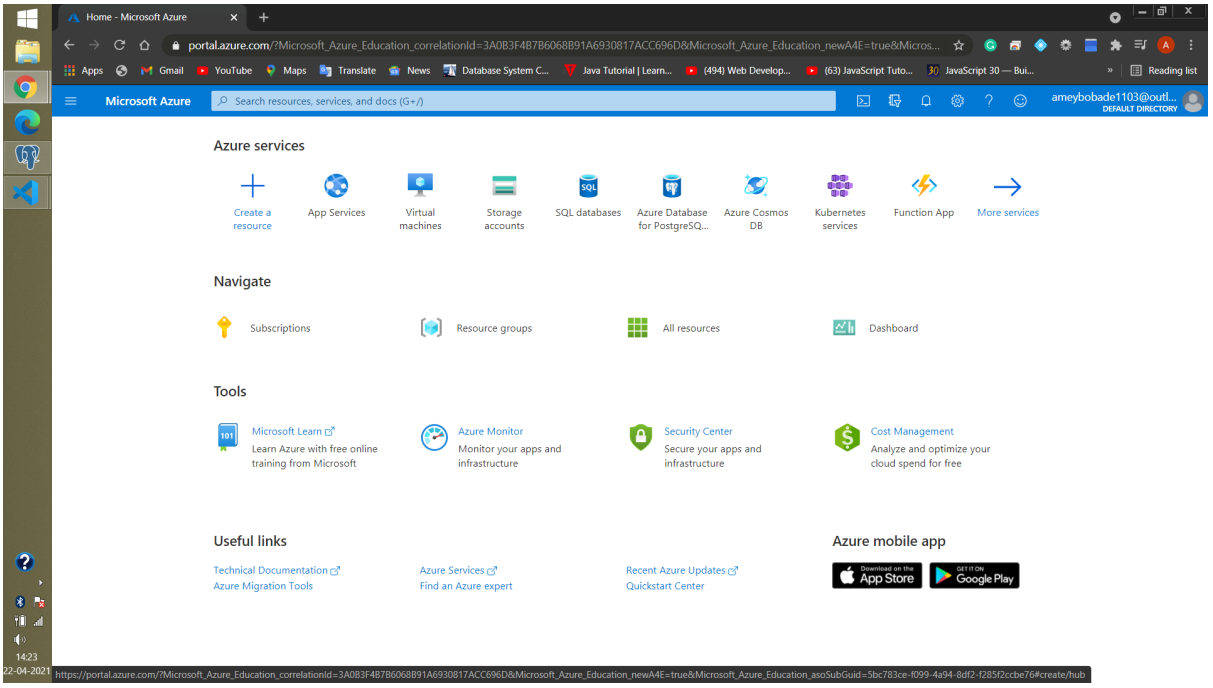
Firebase is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google's infrastructure. **Firebase** is categorized as a NoSQL database program, which stores data in JSON-like documents.

3. Custom app using AMAZON HONEYCODE

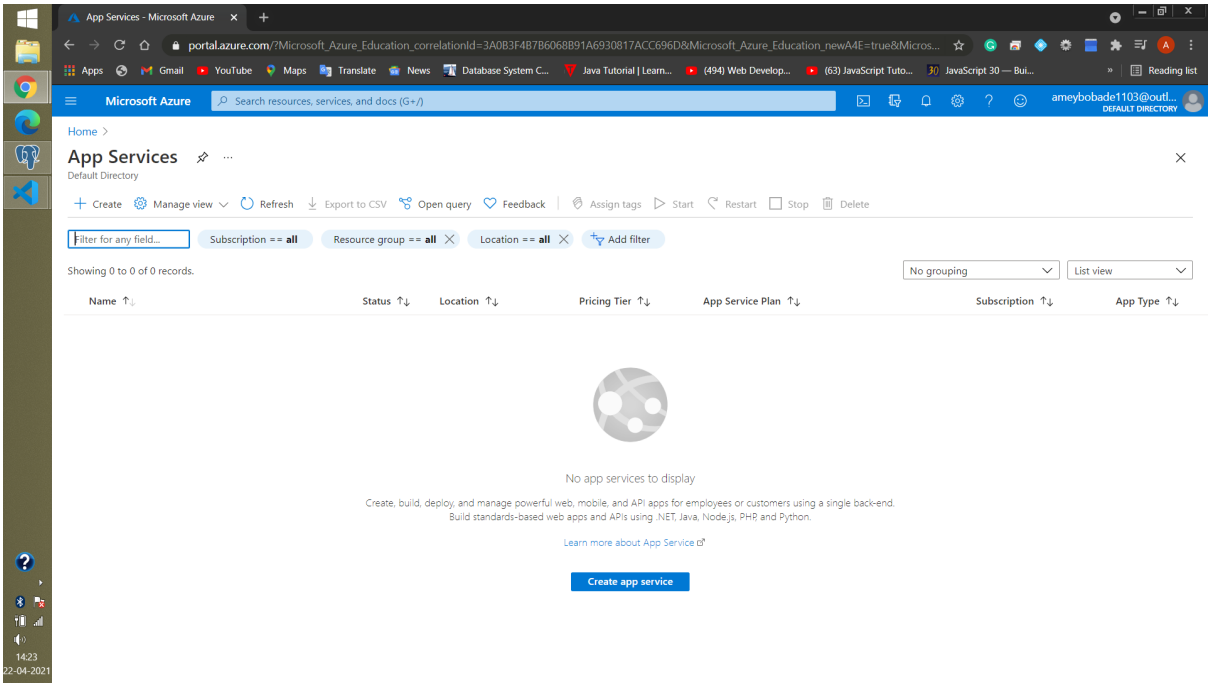
Amazon Honeycode is a fully managed service that allows you to quickly build mobile and web apps for teams—without programming. Build **Amazon Honeycode** apps for managing almost anything, like projects, customers, operations, approvals, resources, and even your team.

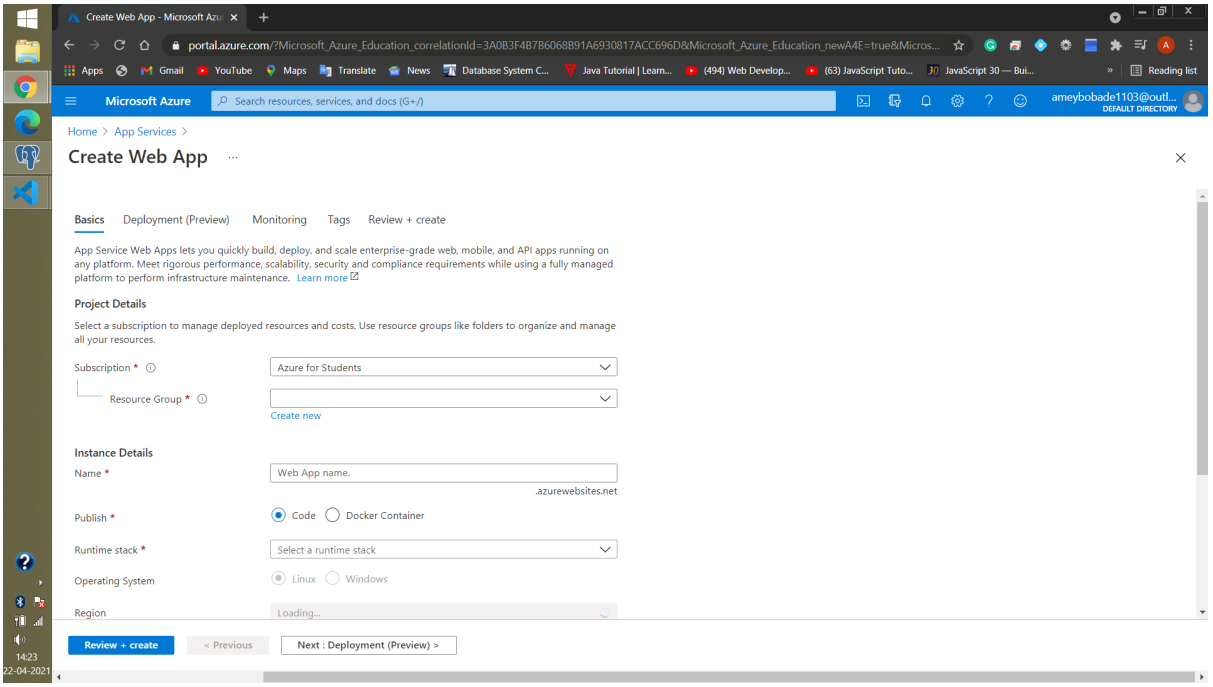
1.Hosting a website using Microsoft Azure

This is the Azure portal where we can view and manage all of your applications in one unified hub—including web apps, databases, virtual machines, virtual networks, storage, and Visual Studio team projects. Enjoy the flexibility of using the Azure portal's graphical experience or the integrated command-line experience provided by Cloud Shell.



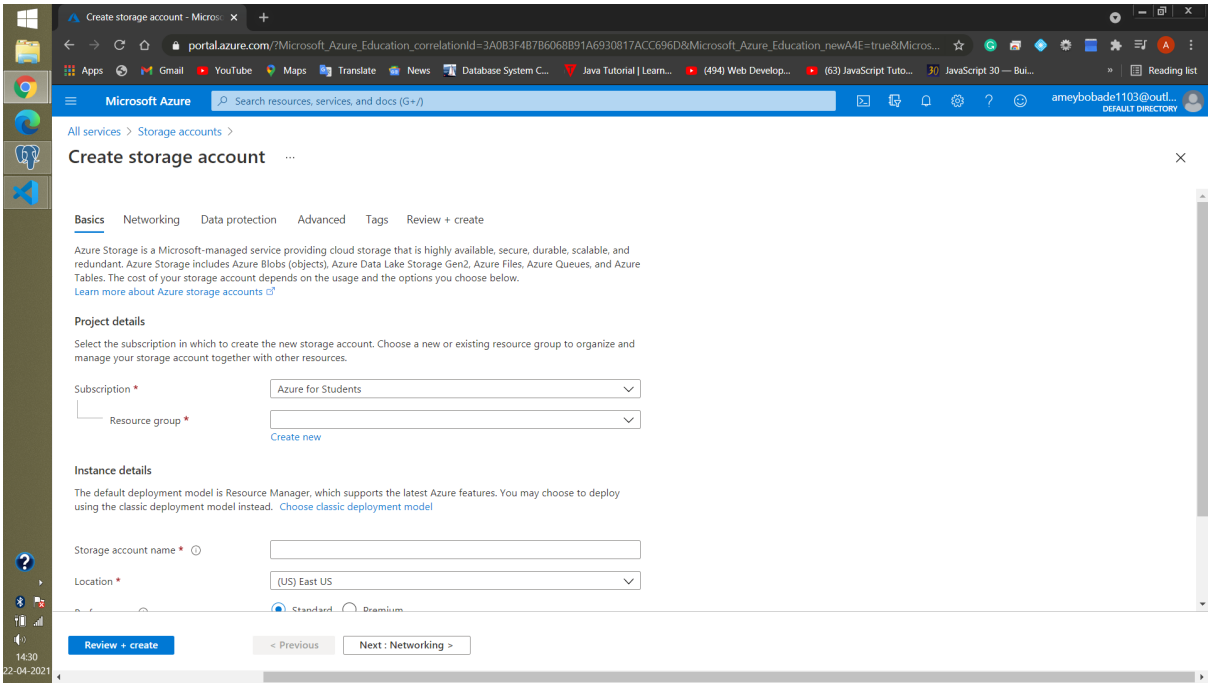
Azure Web Apps is a cloud computing based platform for hosting websites, created and operated by Microsoft. It is a platform as a service which allows publishing Web apps running on multiple frameworks and written in different programming languages, including Microsoft proprietary ones and 3rd party ones.



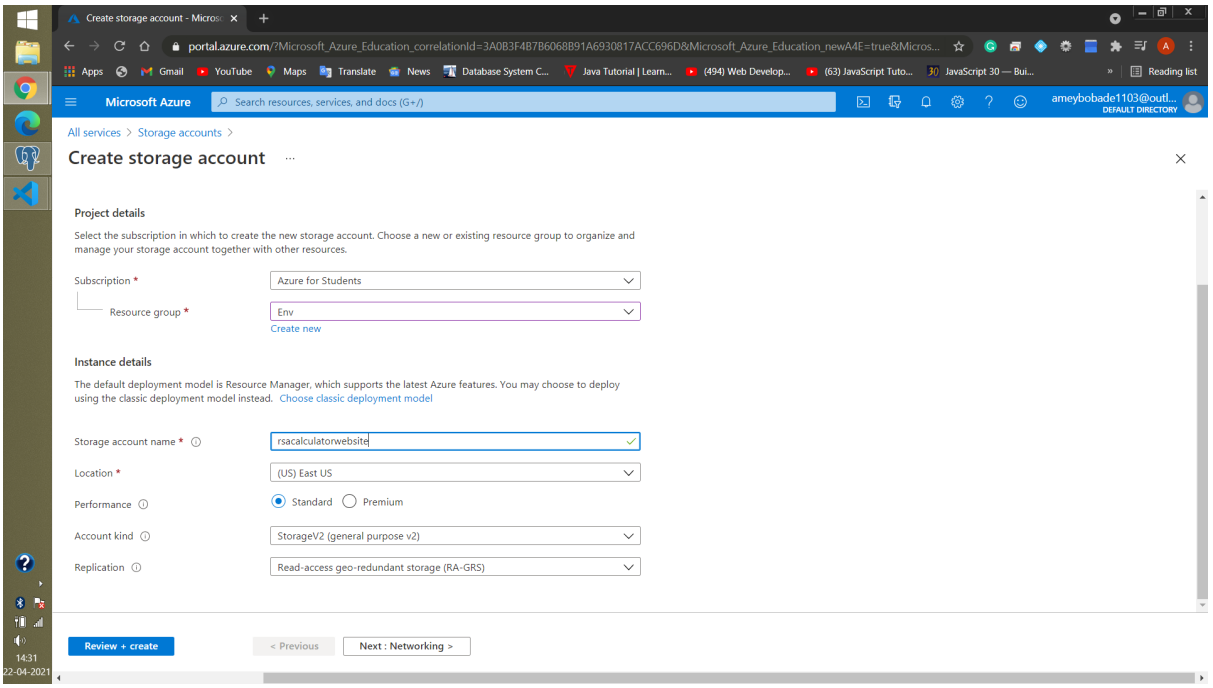


So firstly we'll have to create a azure Storage account. Fill in the given options as we like it. An **Azure storage account** contains all of your **Azure Storage** data objects: blobs, files, queues, tables, and disks. The **storage account** provides a unique namespace for your **Azure Storage** data that is accessible from anywhere in the world over HTTP or HTTPS.

A **resource group** is a container that holds related **resources** for an **Azure** solution. The **resource group** can include all the **resources** for the solution, or only those **resources** that you want to manage as a **group**.

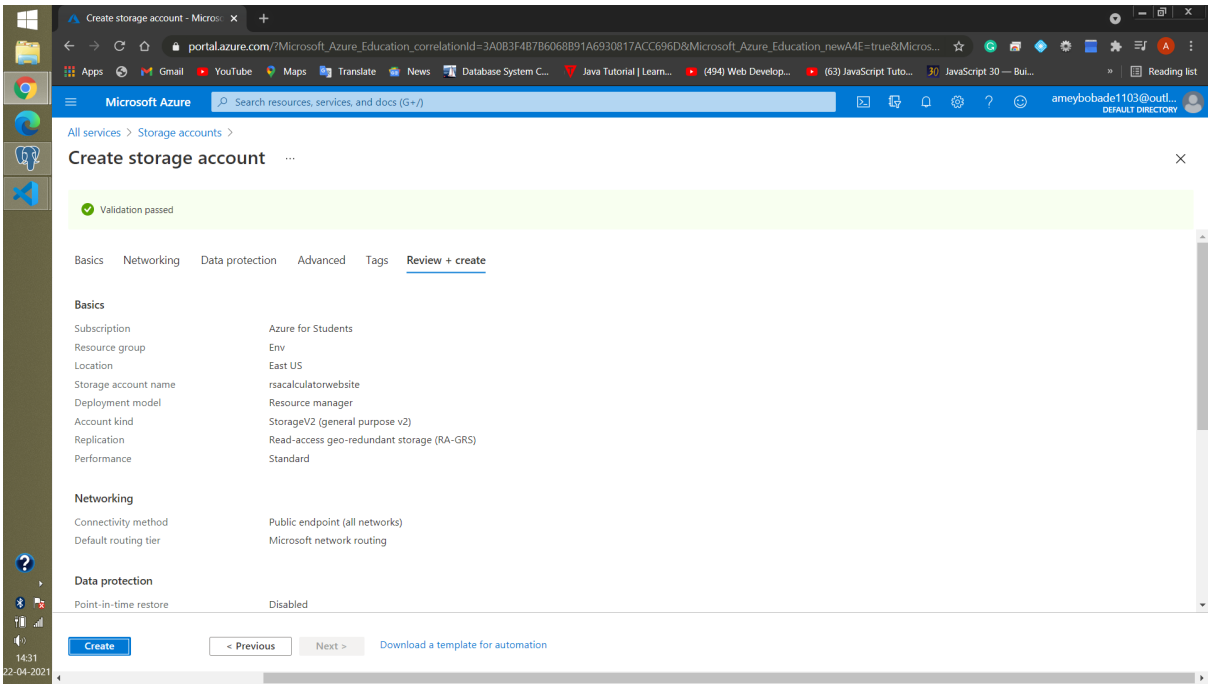


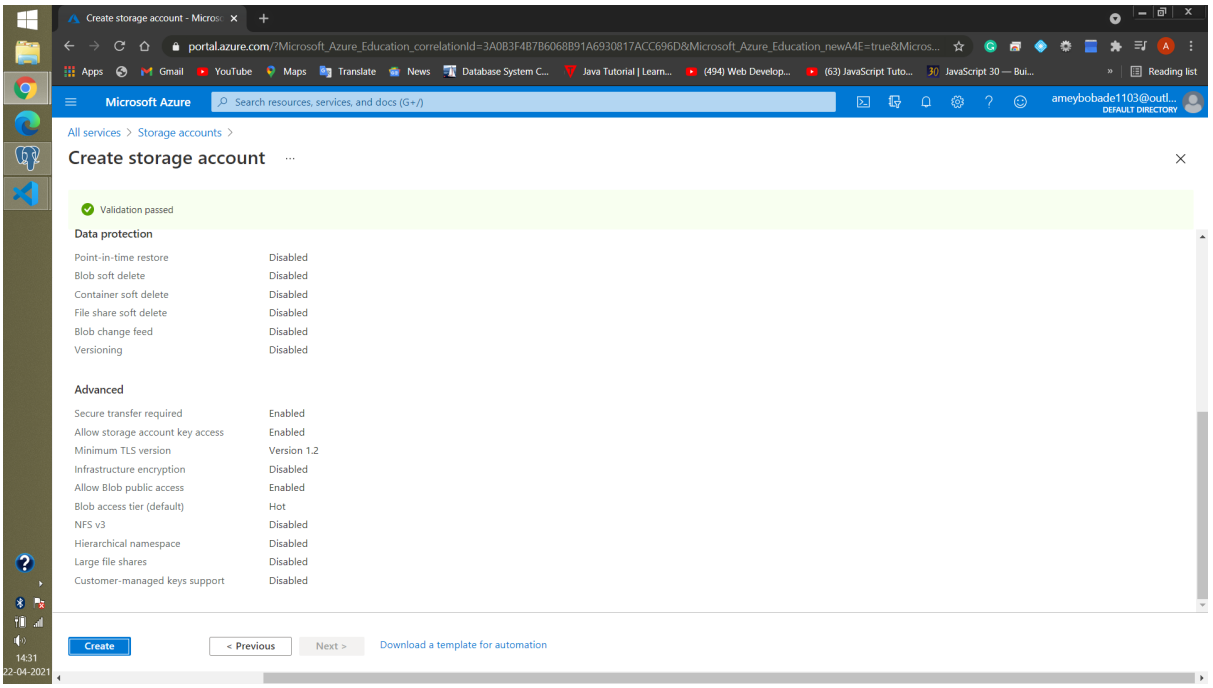
The storage account name has to be unique in the world. A **storage account** provides a unique namespace in **Azure** for your data. Every object that you store in **Azure Storage** has an address that includes your unique **account name**. The combination of the **account name** and the **Azure Storage** service endpoint forms the endpoints for your **storage account**.



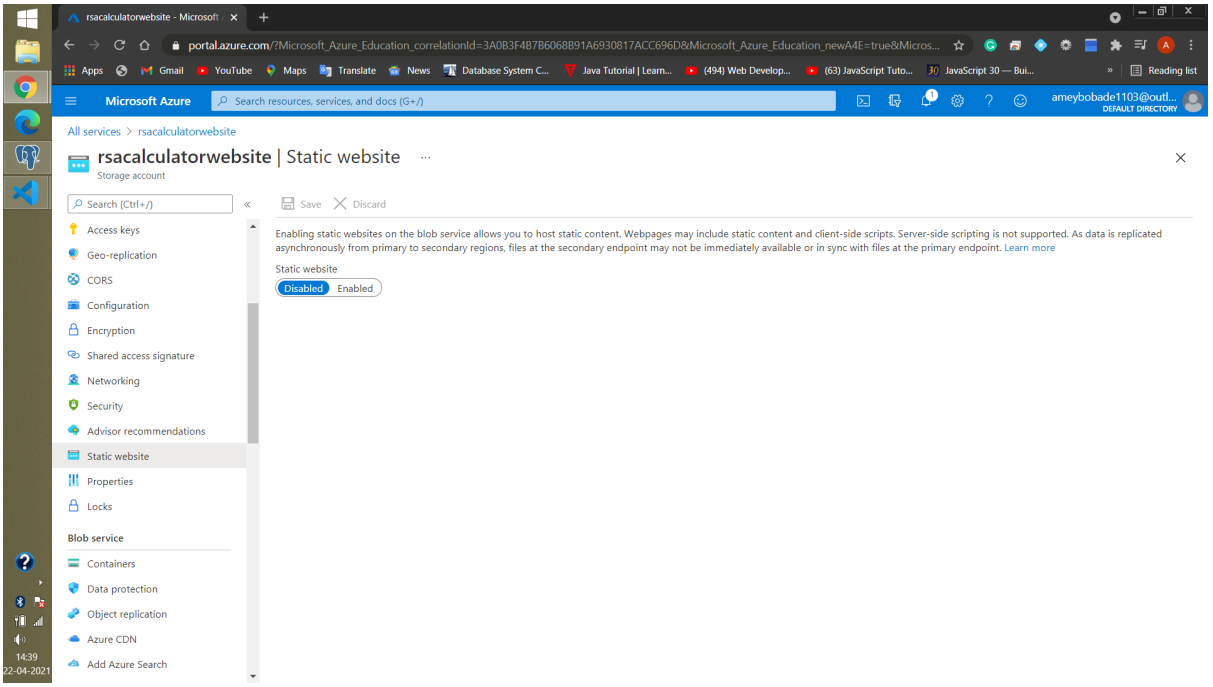
Keep going on next on each tab on default options itself

Now the storage account is created and is ready to be deployed.

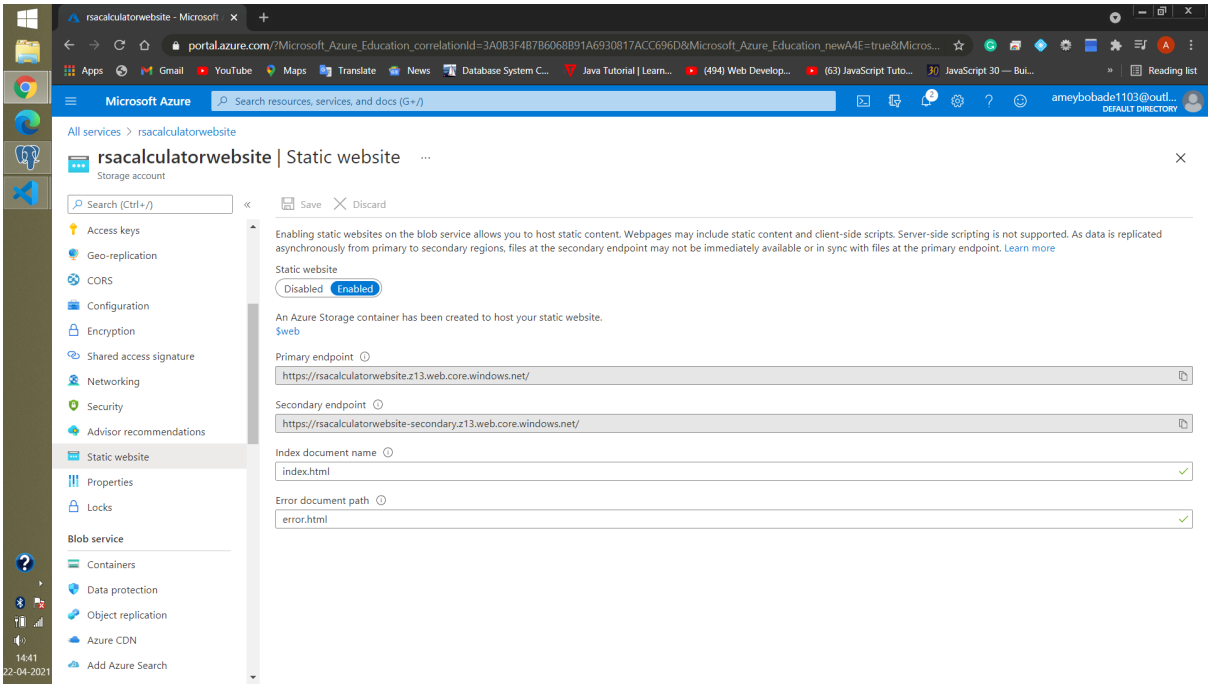




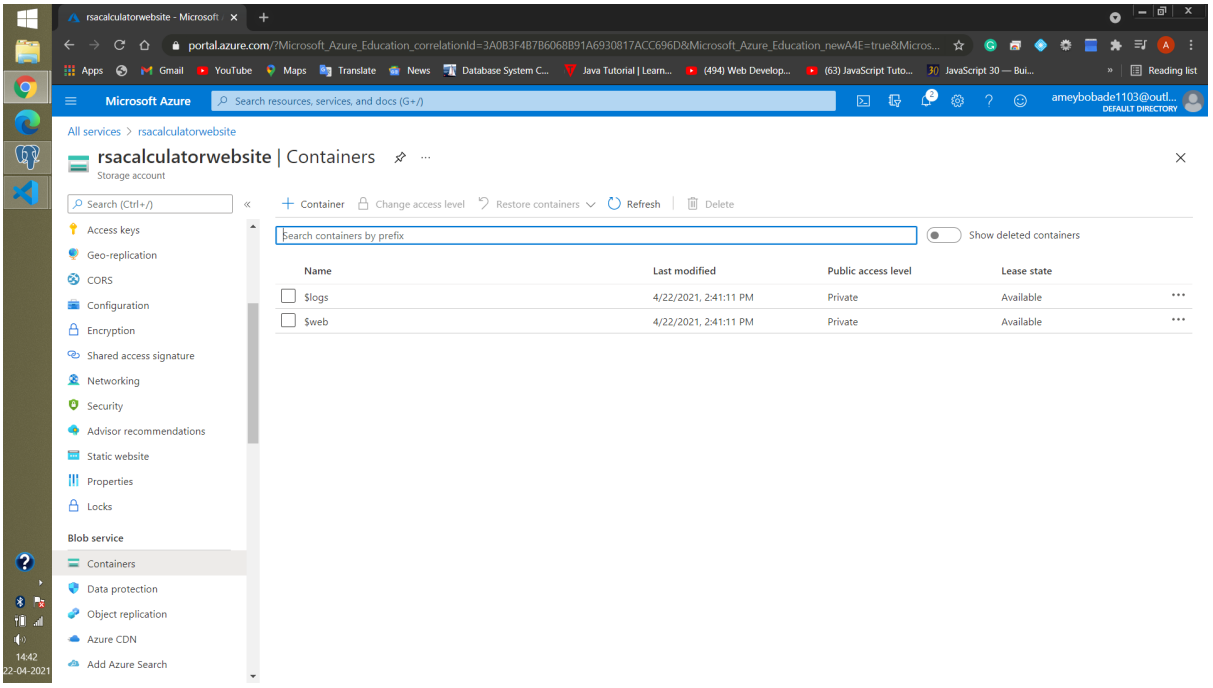
Once this is deployed, we need to enable our static website.



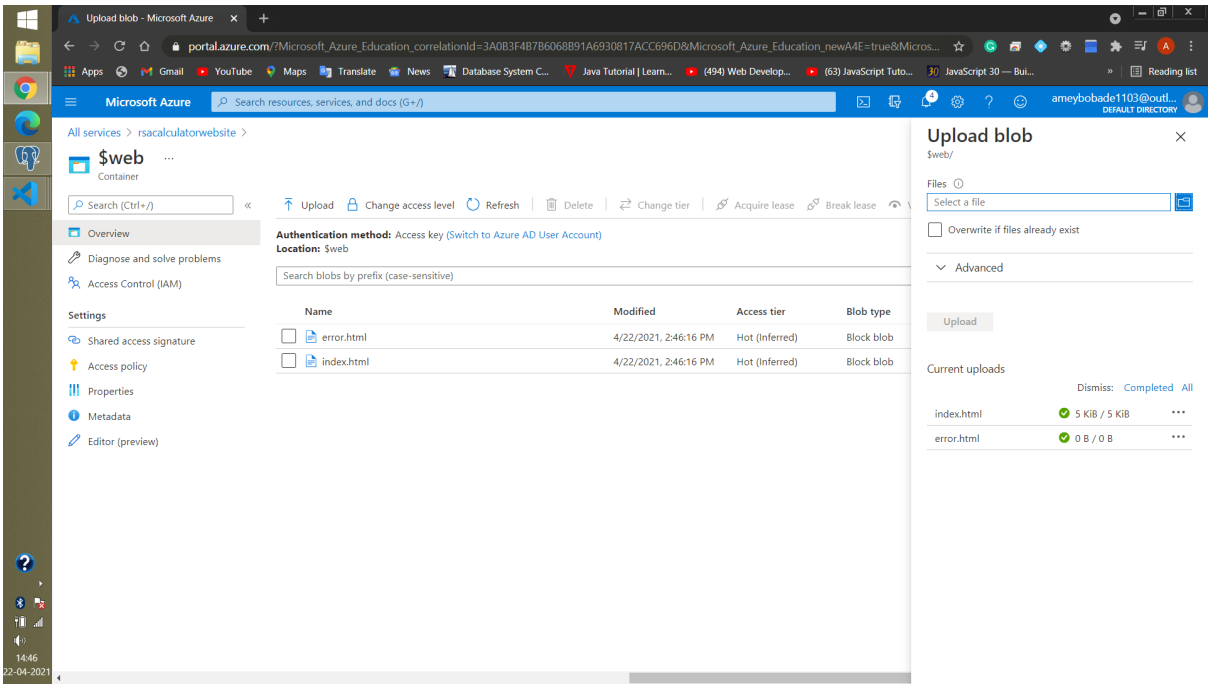
Azure Private **Endpoint** is a network interface that connects you privately and securely to a service powered by **Azure** Private Link. Private **Endpoint** uses a private IP address from your VNet, effectively bringing the service into your VNet.



A **container** is a readymade software environment that has the application code and its dependency's preloaded within an image. ... The **container** engine allows each application (**container**) to run on top of the host operating system but remain isolated from each other.

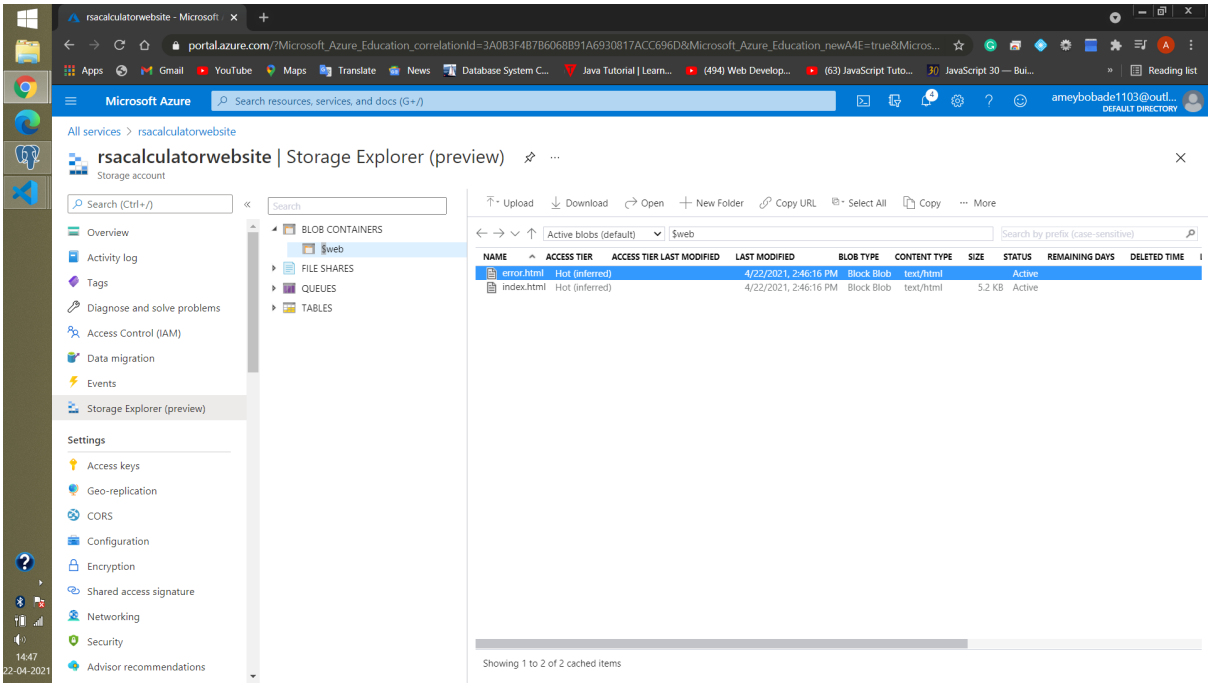


Then you'll see the $web and $logs folder. Open the $web folder and here you'll have to upload your files.



A **blob** is a data type that can store binary data. This is different from most other data types used in databases, such as integers, floating point numbers, characters, and strings, which store letters and numbers. Since **blobs** can store binary data, they can be used to store images or other multimedia files.

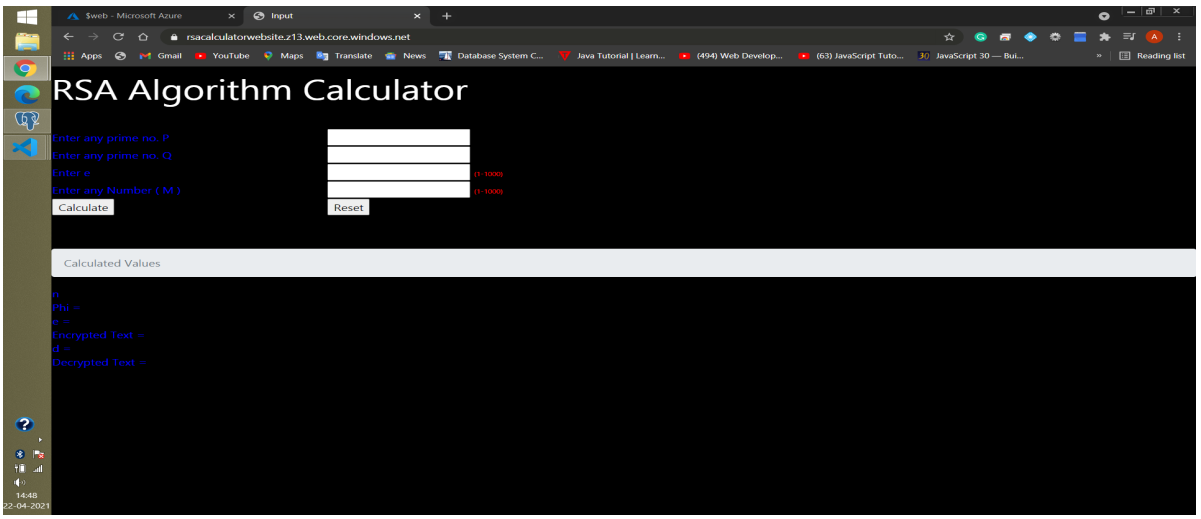
Then open Storage Explorer (preview). **Azure Storage Explorer** is an application which helps you to easily access the **Azure storage** account through any device on any platform, be it Windows, MacOS, or Linux. You can easily connect to your subscription and manipulate your tables, blobs, queues, and files.



Now your website is deployed and ready to go!

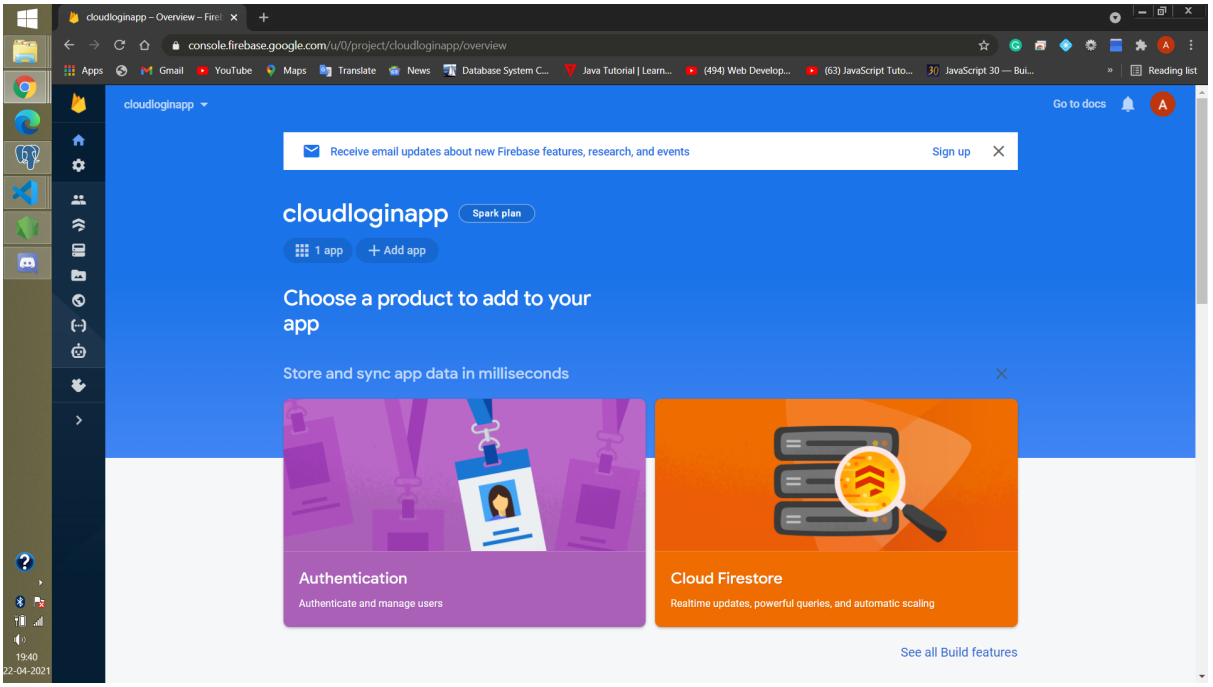
https://rsacalculatorwebsite.z13.web.core.windows.net/

RSA is a public-key cryptosystem and is widely used for secure data transmission. In such a cryptosystem, the encryption key is public and it is different from the decryption key which is kept secret (private).

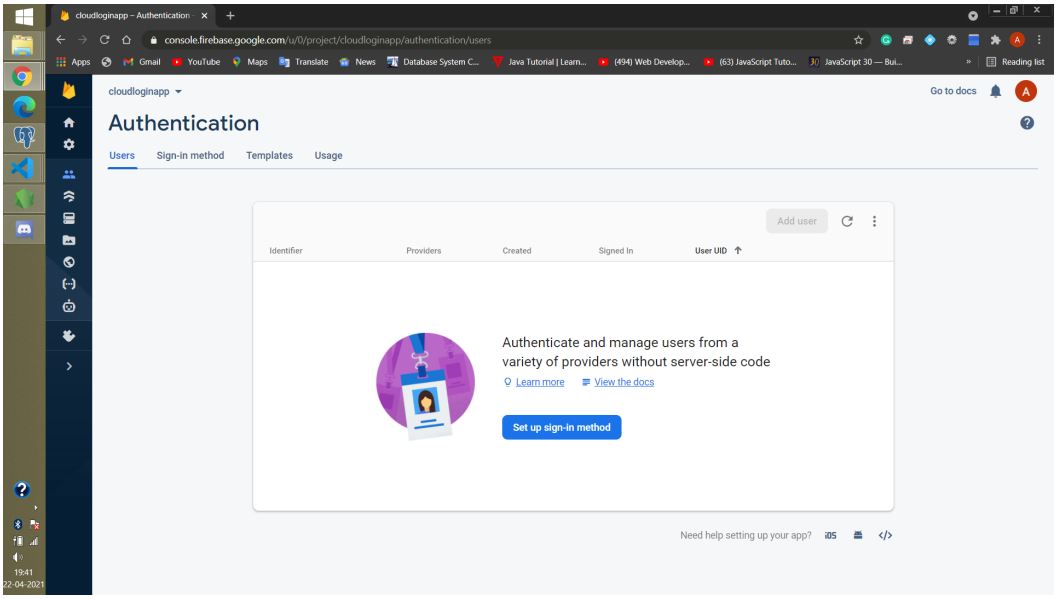


2.GOOGLE FIREBASE AUTHENTICATION

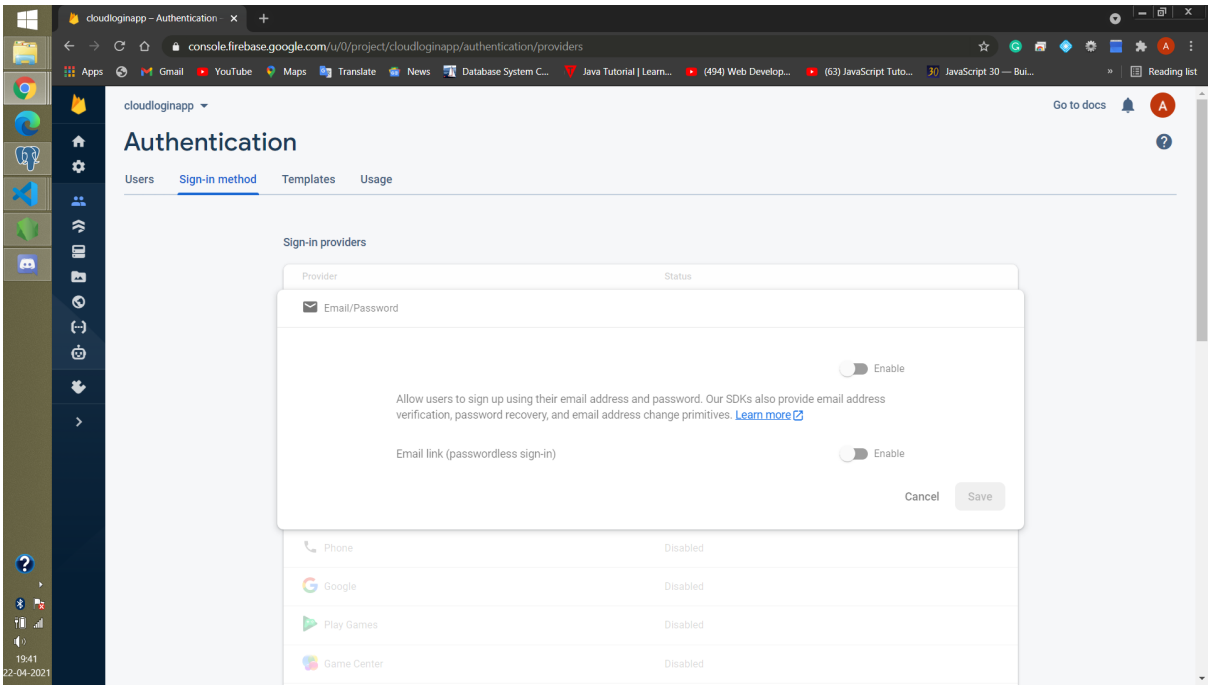
Step1:Login to your firebase account and create a new app. Then, choose any one product for your app mentioned below.Here, we are using authentication. **Firebase Authentication** provides backend services, easy-to-use SDKs, and ready-made UI libraries to **authenticate** users to your app. It supports **authentication** using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.



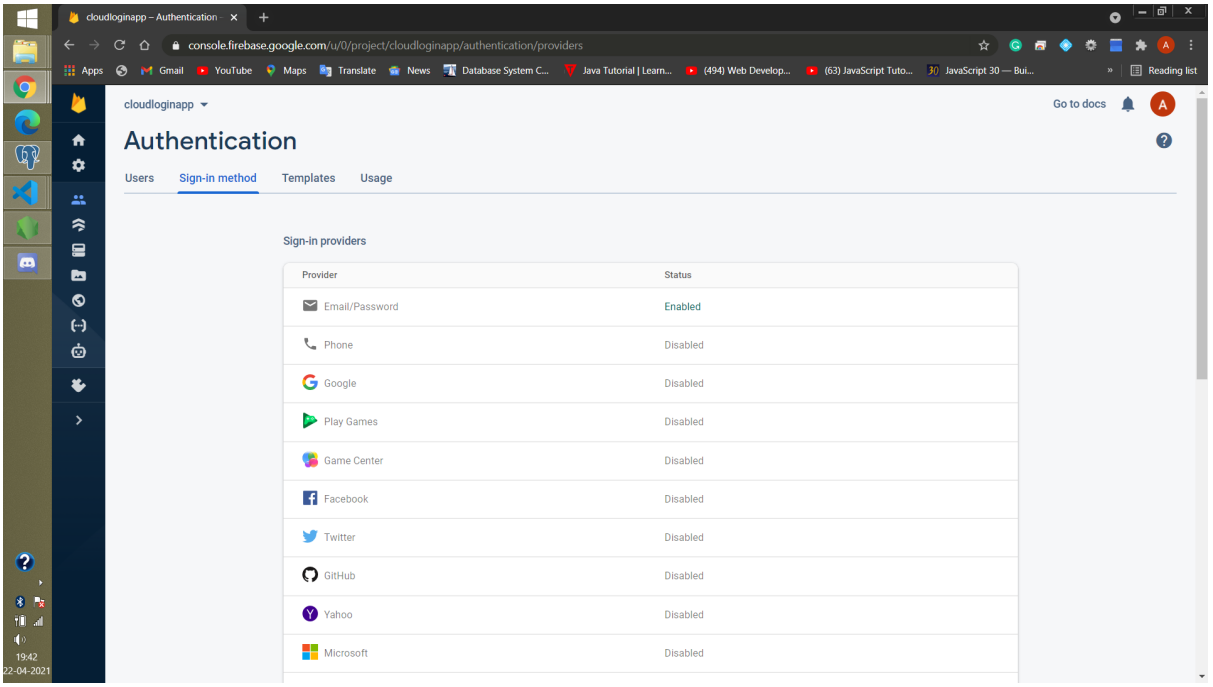
Step 2: Now , setting up sign in method



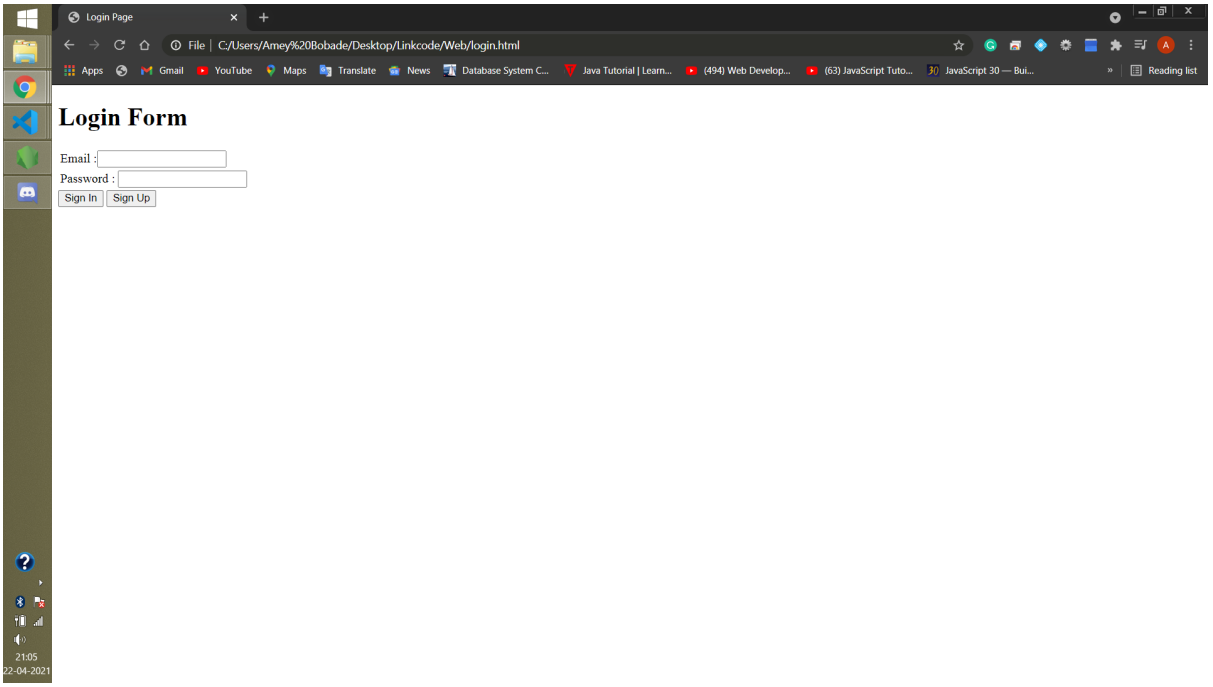
Step 3:Here we are allowing users to sign-up using their email address and password by enabling the service mentioned below.



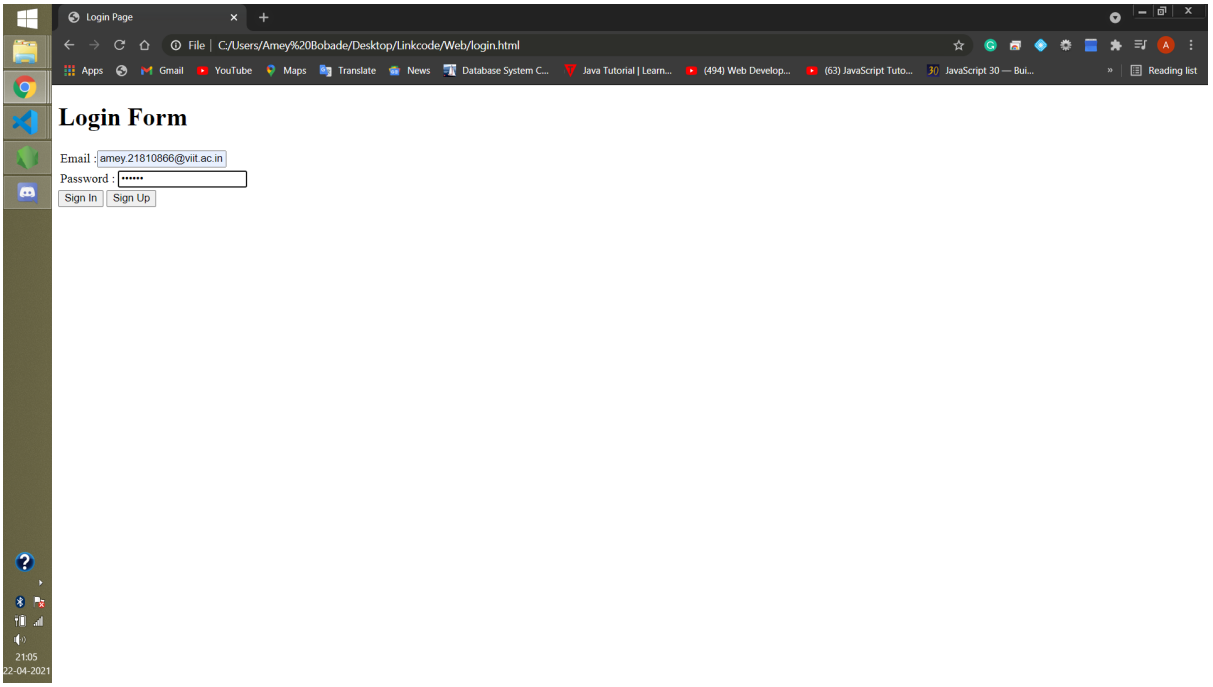
Enabling the feature->



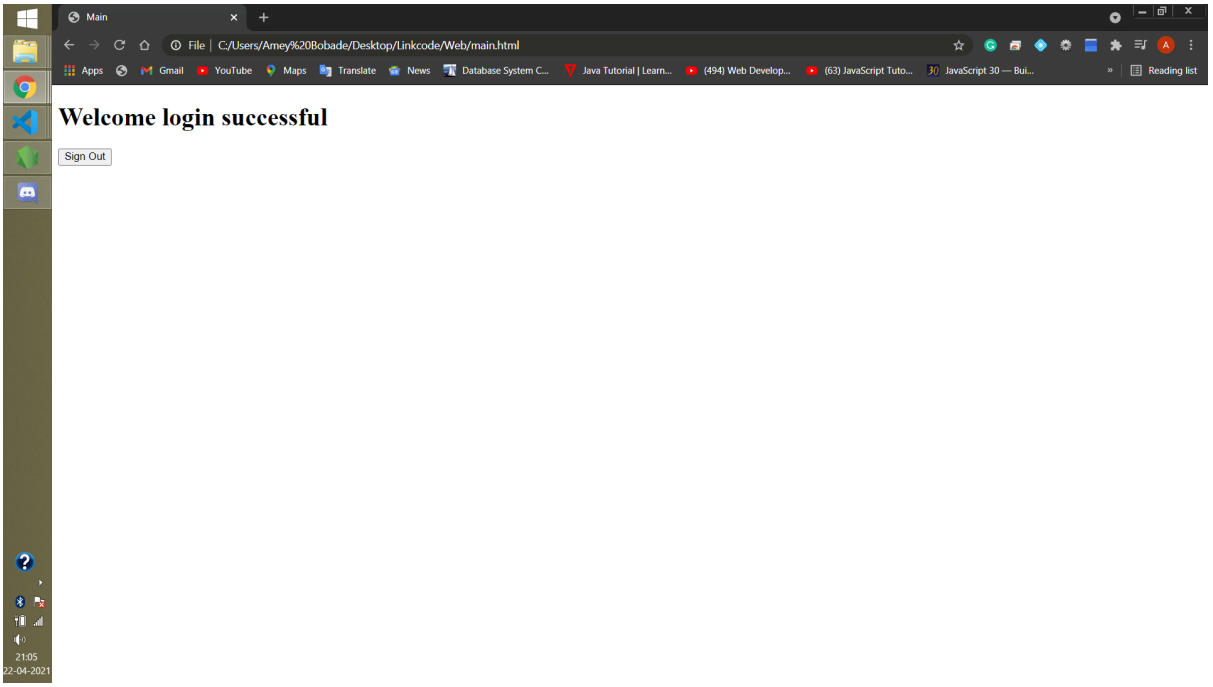
Here is the login form created.



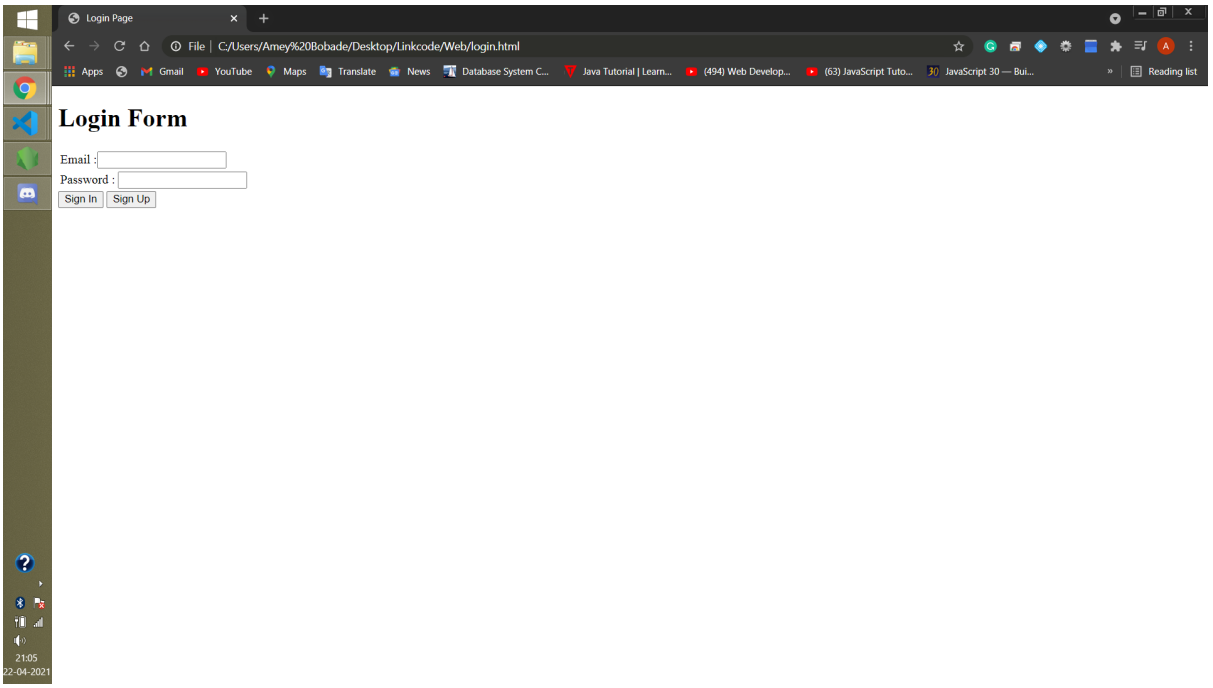
Here, we are trying to Sign-in through a mail and password.



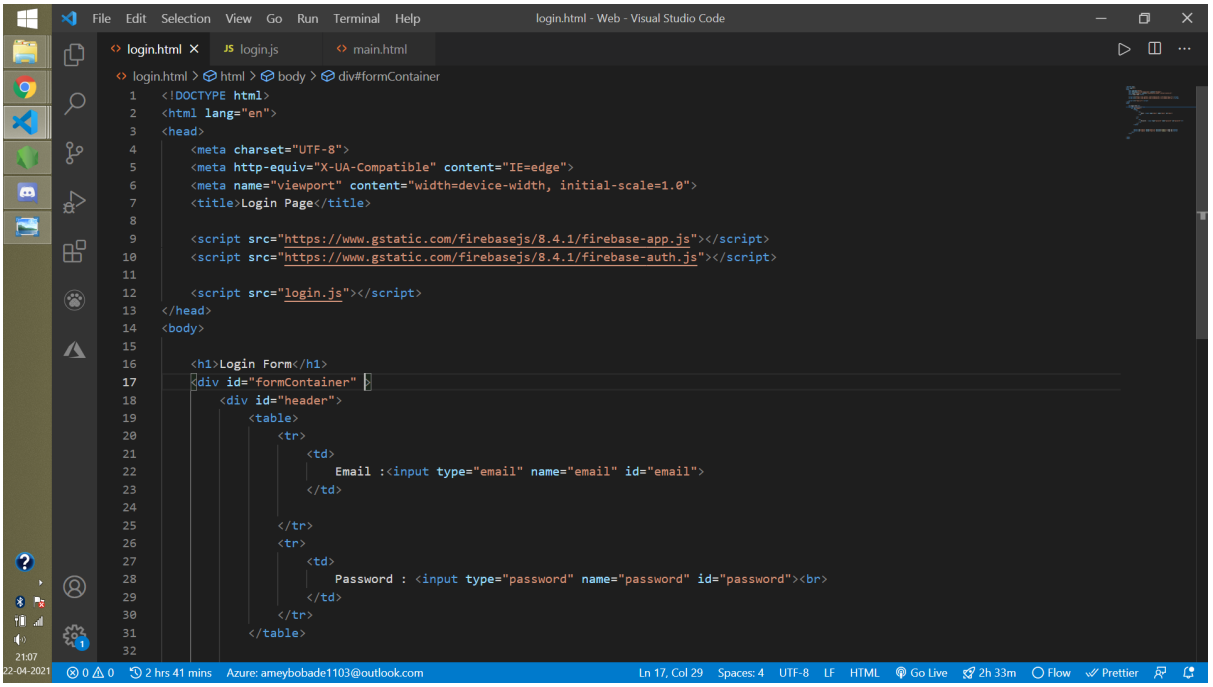
As you can see, login has been successful.



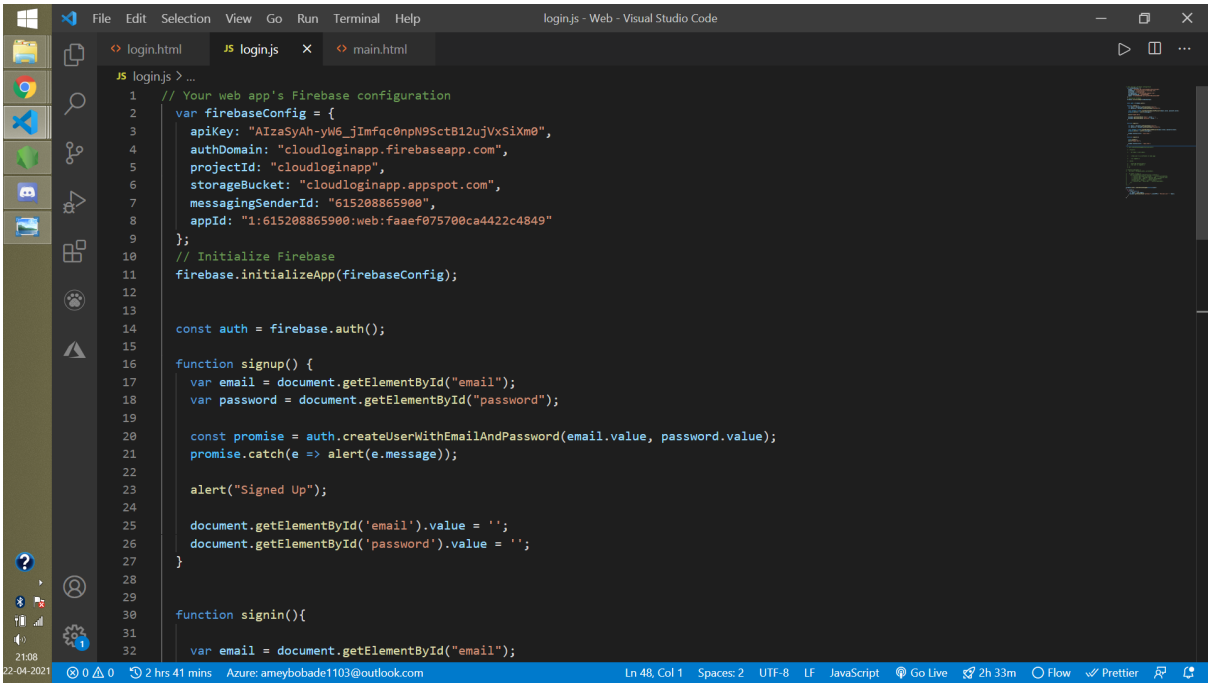
You can also sign-out.

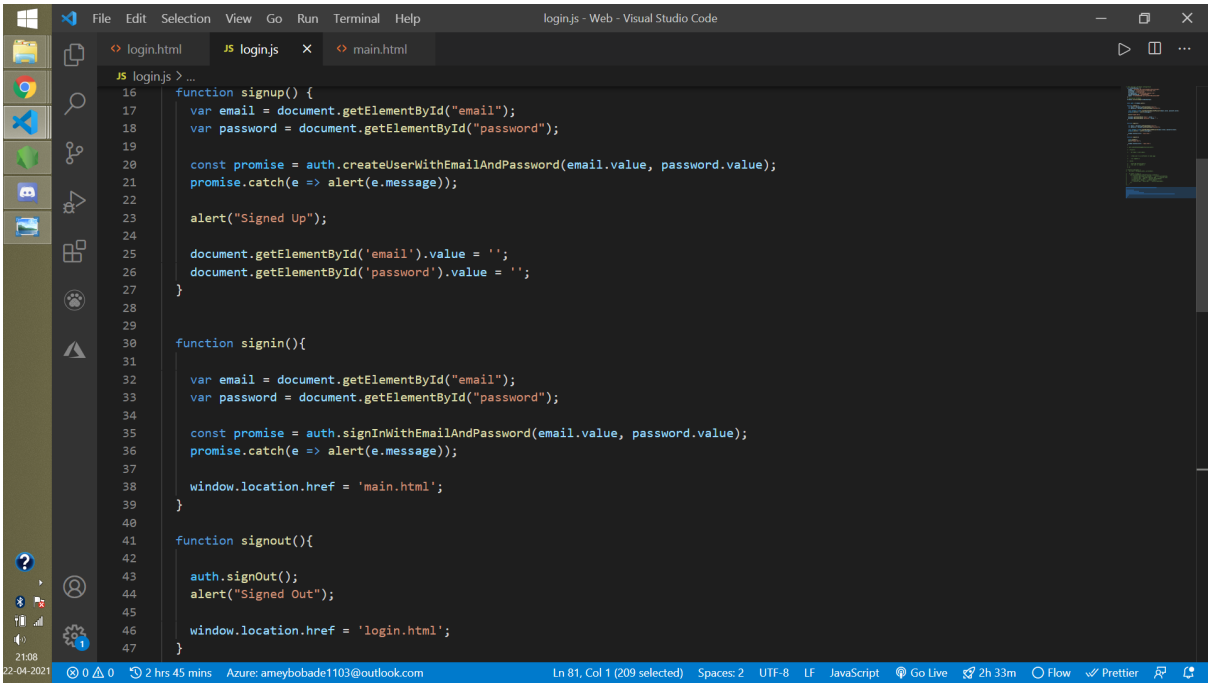


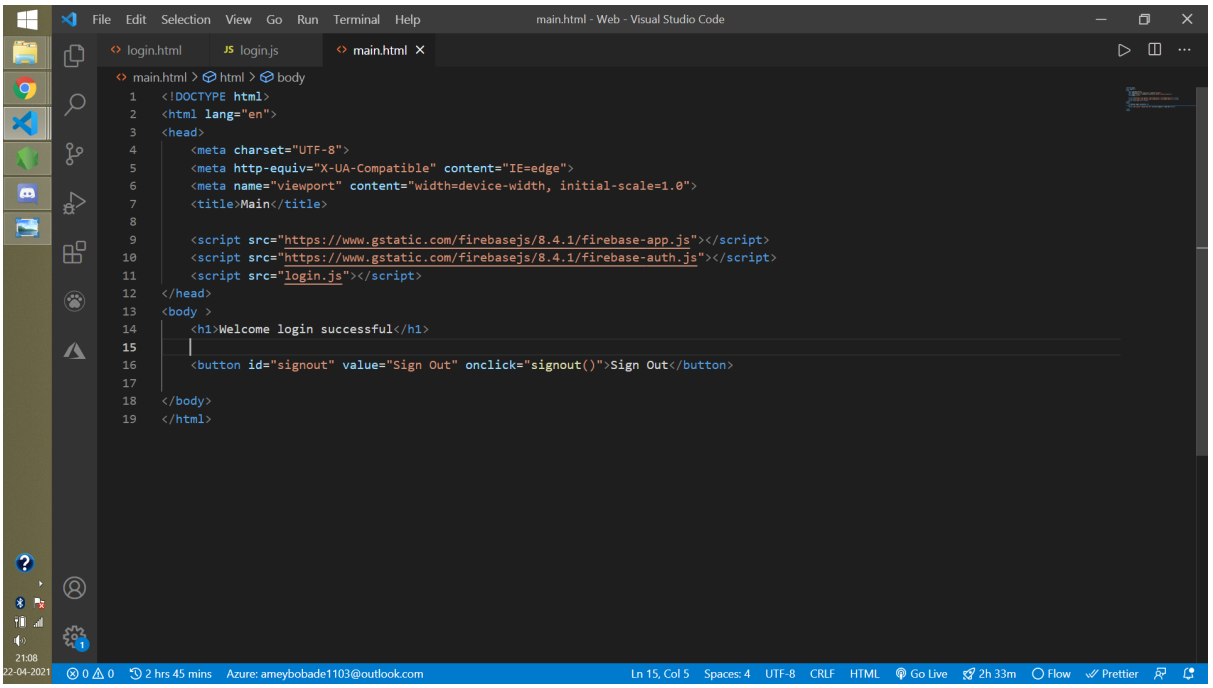
Here is the screenshot of the code we used for authentication.


File Edit Selection View Go Run Terminal Help
login.html - Web - Visual Studio Code
login.html
login.js
main.html
login.html > html > body > div#formContainer
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Login Page</title>
<script src="https://www.gstatic.com/firebasejs/8.4.1/firebase-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/8.4.1/firebase-auth.js"></script>
<script src="login.js"></script>
</head>
<body>
<h1>Login Form</h1>

<table>
<tr>
<td>
Email :<input type="email" name="email" id="email">
</td>
</tr>
<tr>
<td>
Password : <input type="password" name="password" id="password">
</td>
</tr>
</table>
21:07
22-04-2021
0 0
2 hrs 41 mins
Azure: ameybobade1103@outlook.com
Ln 17, Col 29
Spaces: 4
UTF-8
LF
HTML
Go Live
2h 33m
Flow
Prettier


File Edit Selection View Go Run Terminal Help
login.js - Web - Visual Studio Code
login.html
login.js
main.html
login.js > ...
// Your web app's Firebase configuration
var firebaseConfig = {
apiKey: "AIzaSyAh-yW6_jImfqc0npN9SctB12ujVxSiXm0",
authDomain: "cloudloginapp.firebaseapp.com",
projectId: "cloudloginapp",
storageBucket: "cloudloginapp.appspot.com",
messagingSenderId: "615208865900",
appId: "1:615208865900:web:faaef075700ca4422c4849"
};
// Initialize Firebase
firebase.initializeApp(firebaseConfig);
const auth = firebase.auth();
function signup() {
var email = document.getElementById("email");
var password = document.getElementById("password");
const promise = auth.createUserWithEmailAndPassword(email.value, password.value);
promise.catch(e => alert(e.message));
alert("Signed Up");
document.getElementById('email').value = '';
document.getElementById('password').value = '';
}
function signin(){
var email = document.getElementById("email");
21:08
22-04-2021
0 0
2 hrs 41 mins
Azure: ameybobade1103@outlook.com
Ln 48, Col 1
Spaces: 2
UTF-8
LF
JavaScript
Go Live
2h 33m
Flow
Prettier


File Edit Selection View Go Run Terminal Help
login.js - Web - Visual Studio Code
login.html
login.js
main.html
login.js > ...
function signup() {
var email = document.getElementById("email");
var password = document.getElementById("password");
const promise = auth.createUserWithEmailAndPassword(email.value, password.value);
promise.catch(e => alert(e.message));
alert("Signed Up");
document.getElementById('email').value = '';
document.getElementById('password').value = '';
}
function signin(){
var email = document.getElementById("email");
var password = document.getElementById("password");
const promise = auth.signInWithEmailAndPassword(email.value, password.value);
promise.catch(e => alert(e.message));
window.location.href = 'main.html';
}
function signout(){
auth.signOut();
alert("Signed Out");
window.location.href = 'login.html';
}
21:08
22-04-2021
0 0 2 hrs 45 mins Azure: ameybobade1103@outlook.com
Ln 81, Col 1 (209 selected) Spaces: 2 UTF-8 LF JavaScript Go Live 2h 33m Flow Prettier


File Edit Selection View Go Run Terminal Help
main.html - Web - Visual Studio Code
login.html
login.js
main.html
main.html > html > body
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Main</title>
<script src="https://www.gstatic.com/firebasejs/8.4.1/firebase-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/8.4.1/firebase-auth.js"></script>
<script src="login.js"></script>
</head>
<body >
<h1>Welcome login successful</h1>
<button id="signout" value="Sign Out" onclick="signout()">Sign Out</button>
</body>
</html>
21:08
22-04-2021
0 0 2 hrs 45 mins Azure: ameybobade1103@outlook.com
Ln 15, Col 5 Spaces: 4 UTF-8 CRLF HTML Go Live 2h 33m Flow Prettier

3.AMAZON HONEYCODE

What Is Amazon Honeycode?

Amazon Honeycode, which is available in beta, is a fully managed service that allows

customers to quickly build powerful mobile and web applications – with no programming

required. Customers who need applications to track and manage things like process

approvals, event scheduling, customer relationship management, user surveys, to-do lists,

and content and inventory tracking no longer need to do so by error-prone methods like

emailing spreadsheets or documents, or hiring and waiting for developers to build costly

custom applications.

With Amazon Honeycode, customers can use a simple visual application builder to create highly

interactive web and mobile applications backed by a powerful AWS-built database to perform tasks like

tracking data over time and notifying users of changes, routing approvals, and facilitating interactive

business processes. Using Amazon Honeycode, customers can create applications that range in

complexity from a task-tracking application for a small team to a project management system that

manages a complex workflow for multiple teams or departments.

Is Amazon Honeycode free?

Amazon Honeycode is free to use for teams with up to 20 members and 2,500 rows per

workbook. Upgrade to a paid plan when you need additional capacity.

Benefits of using Amazon Lex:

With Honeycode, one can create web & mobile apps that can work on a desktop browser,

Android, and iOS platforms.

No programming skills are needed so that anyone can be an app developer.

Honeycode apps can help teams stay in a loop by giving them access to the same data.

Personalized features of Honeycode apps can make teams remain focused. This is possible

because these apps show data that are relevant to each team member.

Honeycode can automate manual steps

Features of Amazon Honeycode:

Customized apps without programming.

Personalize data for each team member.

Workflow automation

Share the app

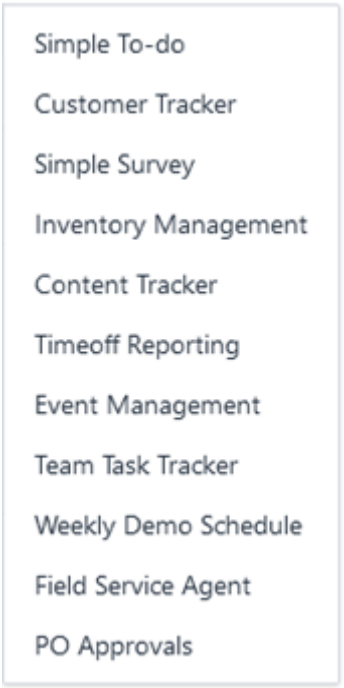
Access anywhere and update anytime.

After having basic knowledge about Amazon Honeycode, let’s learn about how to start

working with it.

How it works:

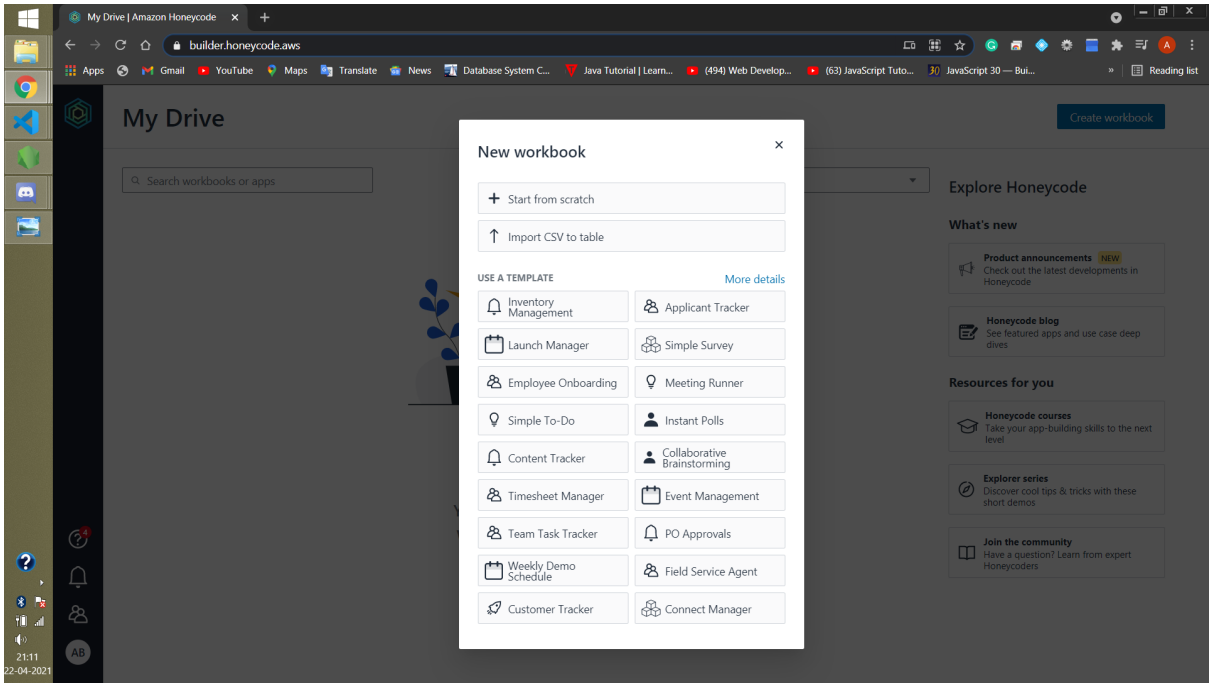
Amazon has built-in templates that can be chosen:



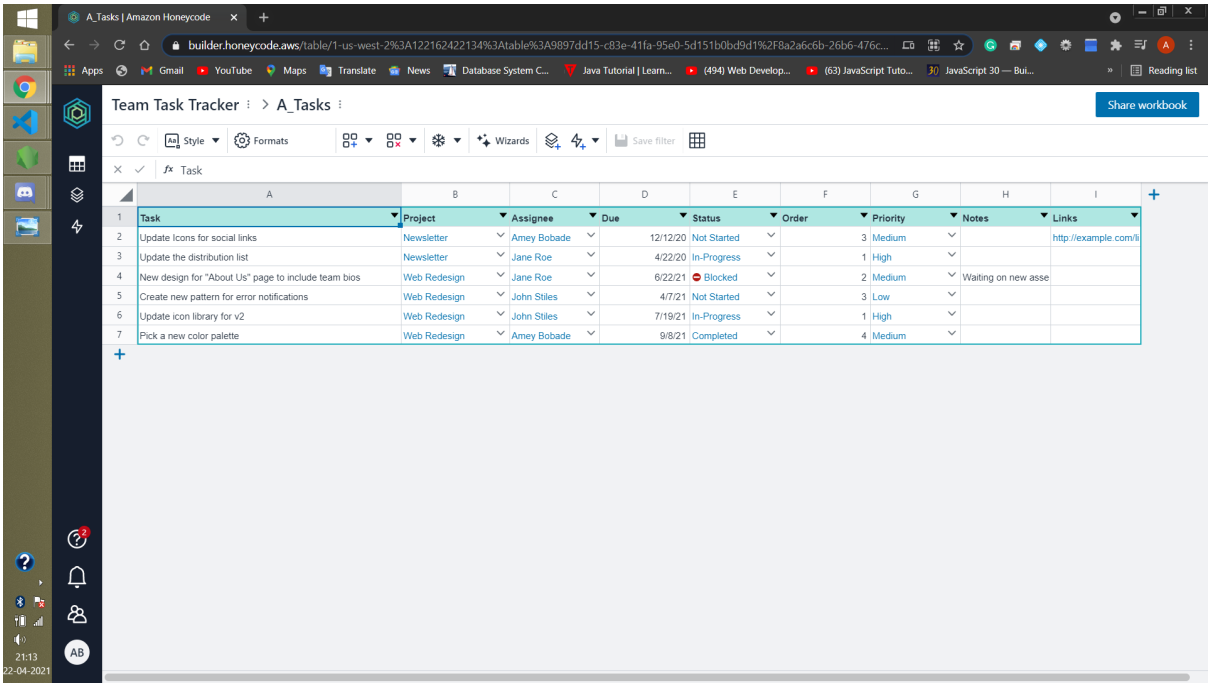
Or you can even build one from scratch with a similar process, we'll go with a built-in template.

Steps:-

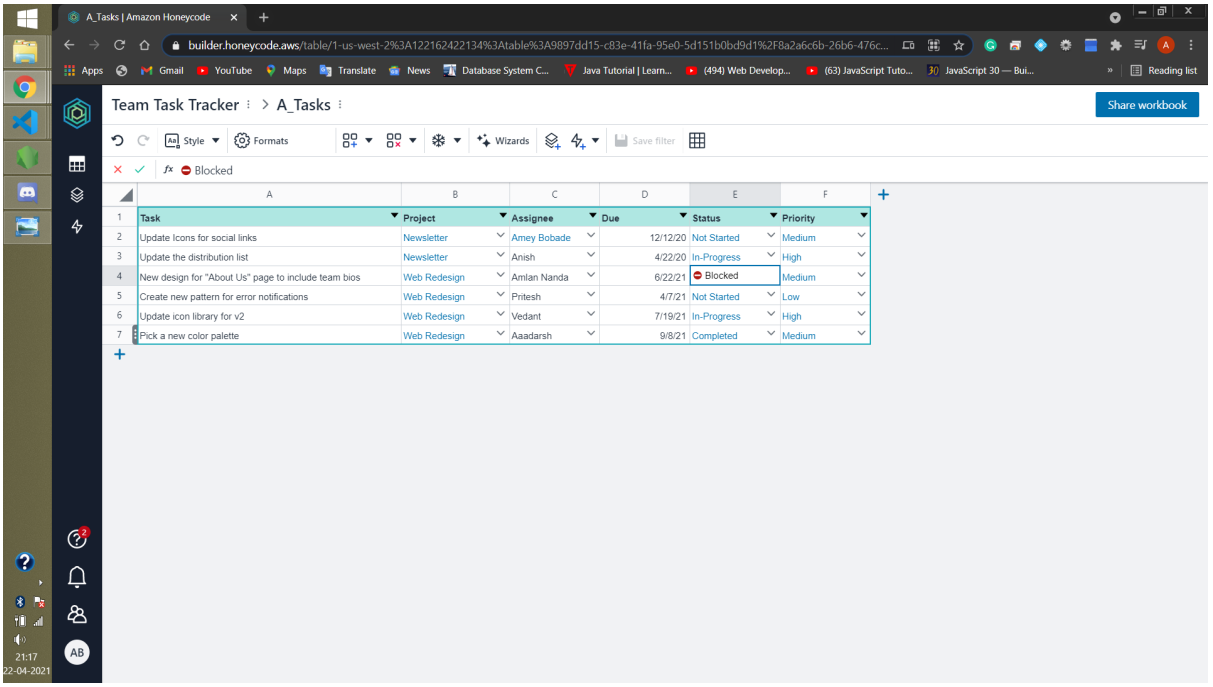
1) GO onto the Amazon Honeycode website, and click on Sign Up.



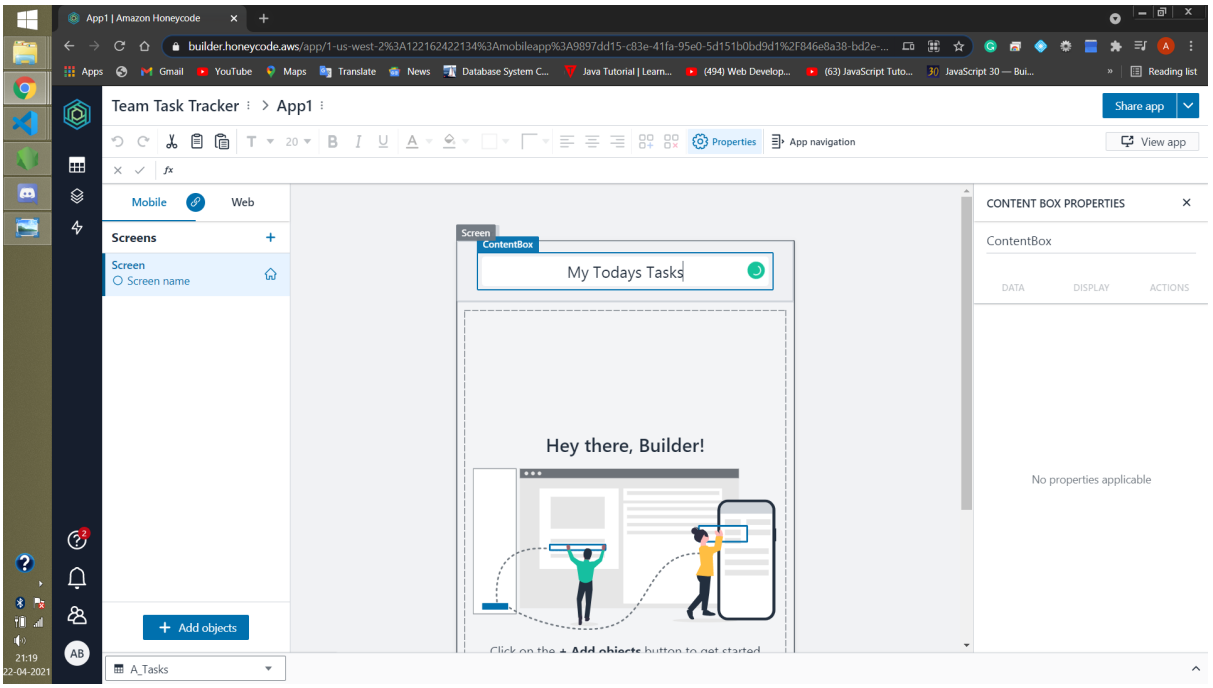
2)Creating an account will lead you to the Honeycode Dashboard



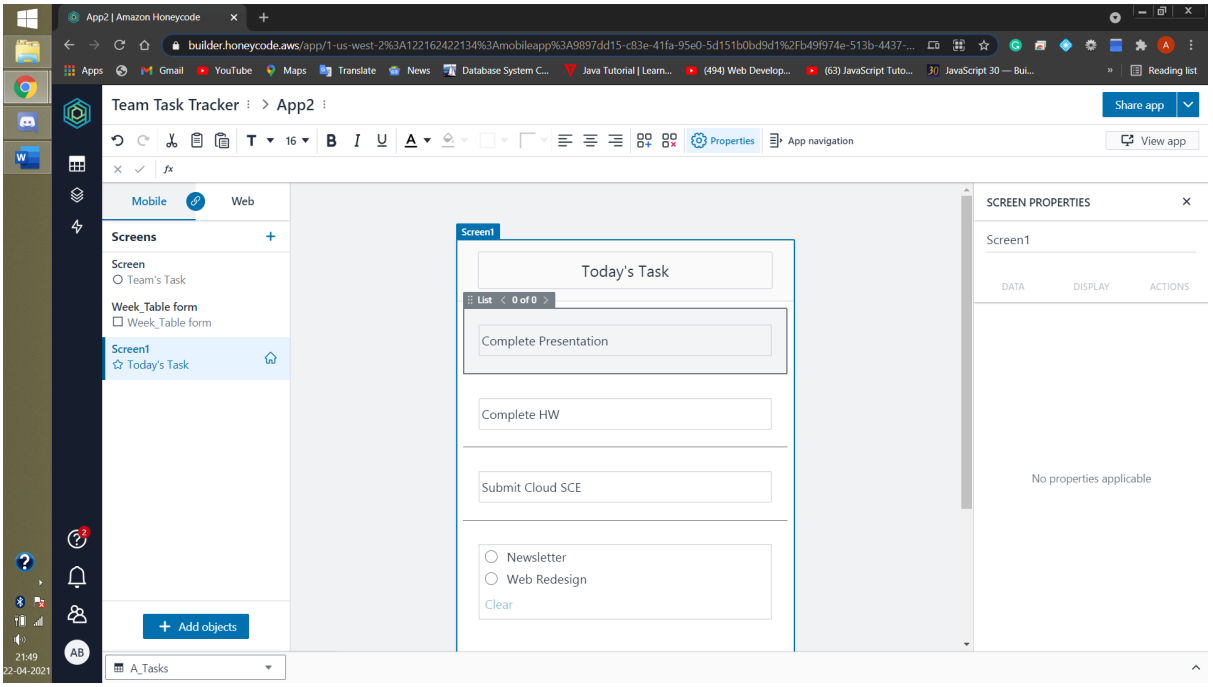
3)Click on Create a New Workbook on Right Top Corner.



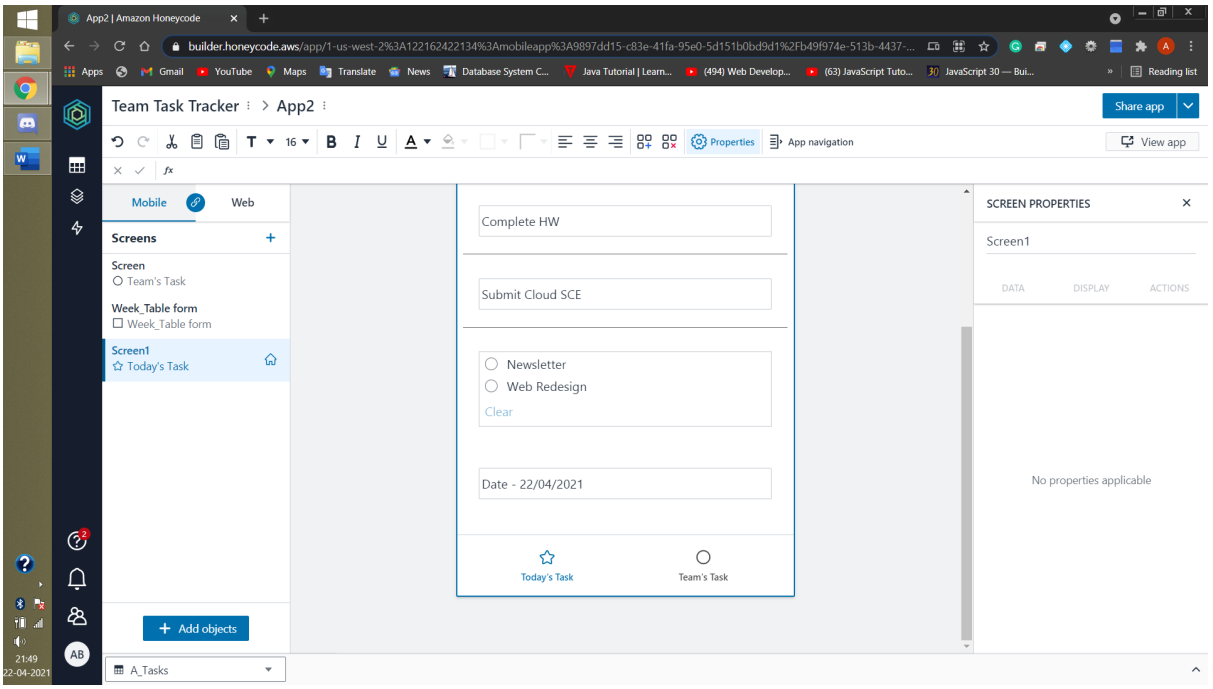
4)Select a Built-In Template for now. We'll go with the To-Do App. We get this spreadsheet for the data in our To-DO App



5) Now, click on the Builder tab on the left panel below the Tables tab, and select ToDo from there.

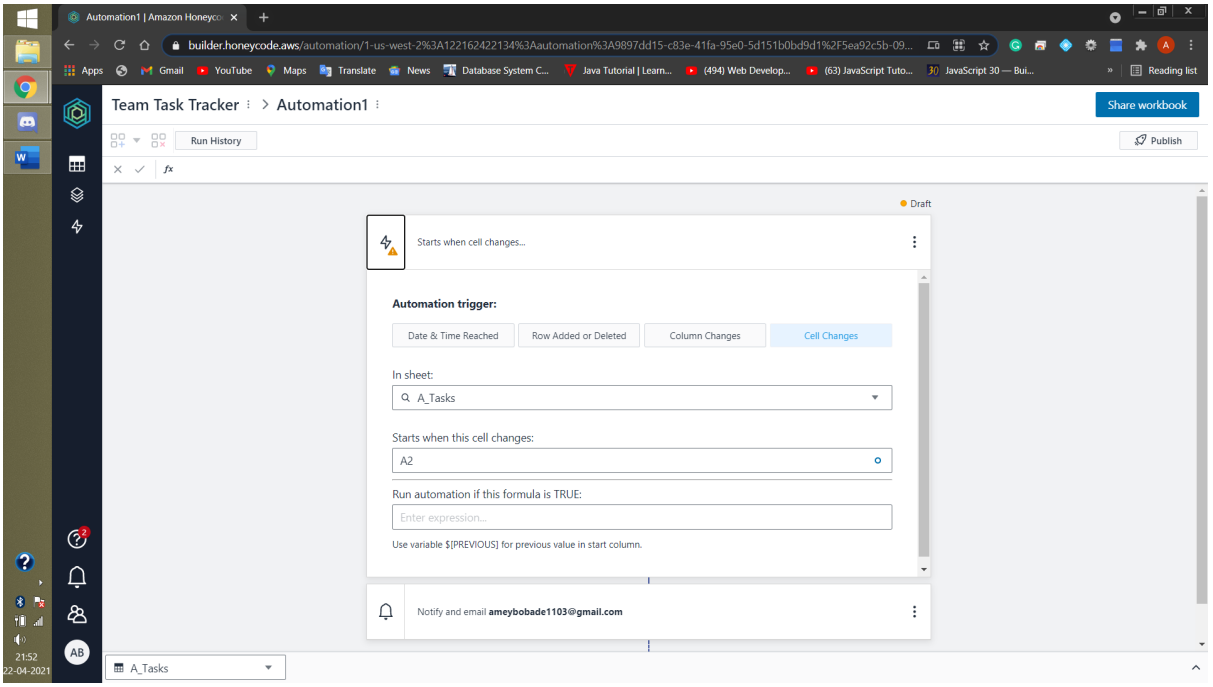


6)Here you can edit anything that should be displayed in your app. From the Bottom Navigation Bar to each and every element present.

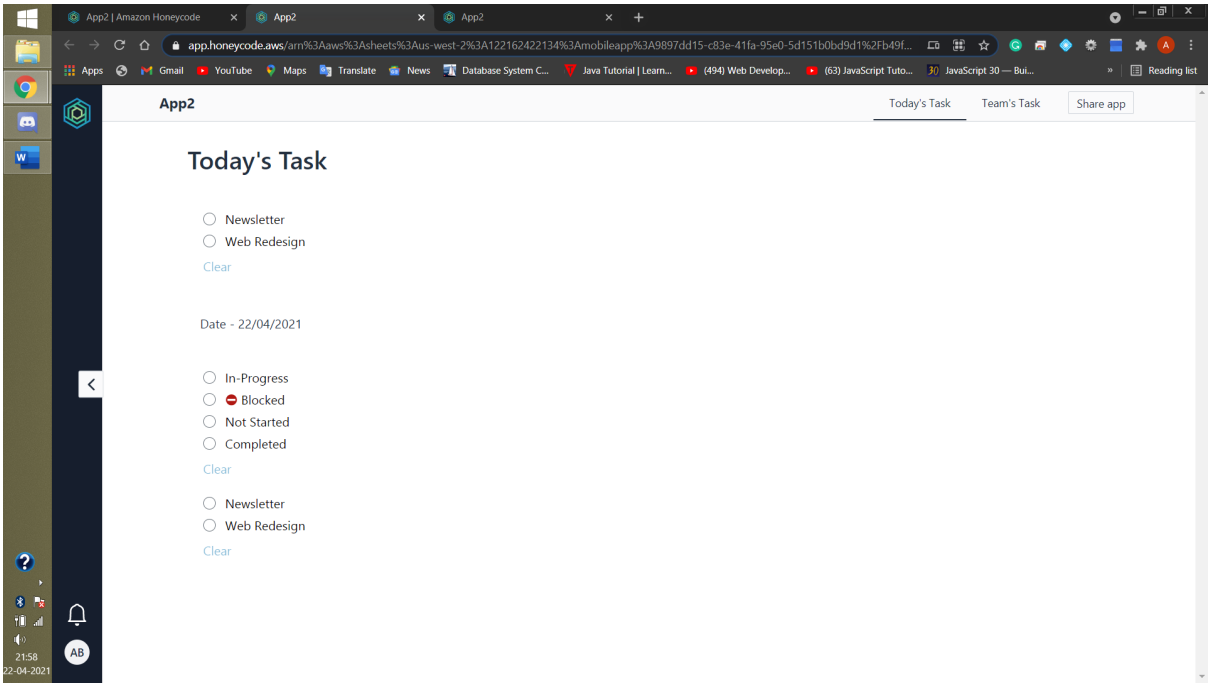


7)We can also add any kind of automations we want from the Automations tab below Builder Tab. Publish the automation from the top-right corner once you

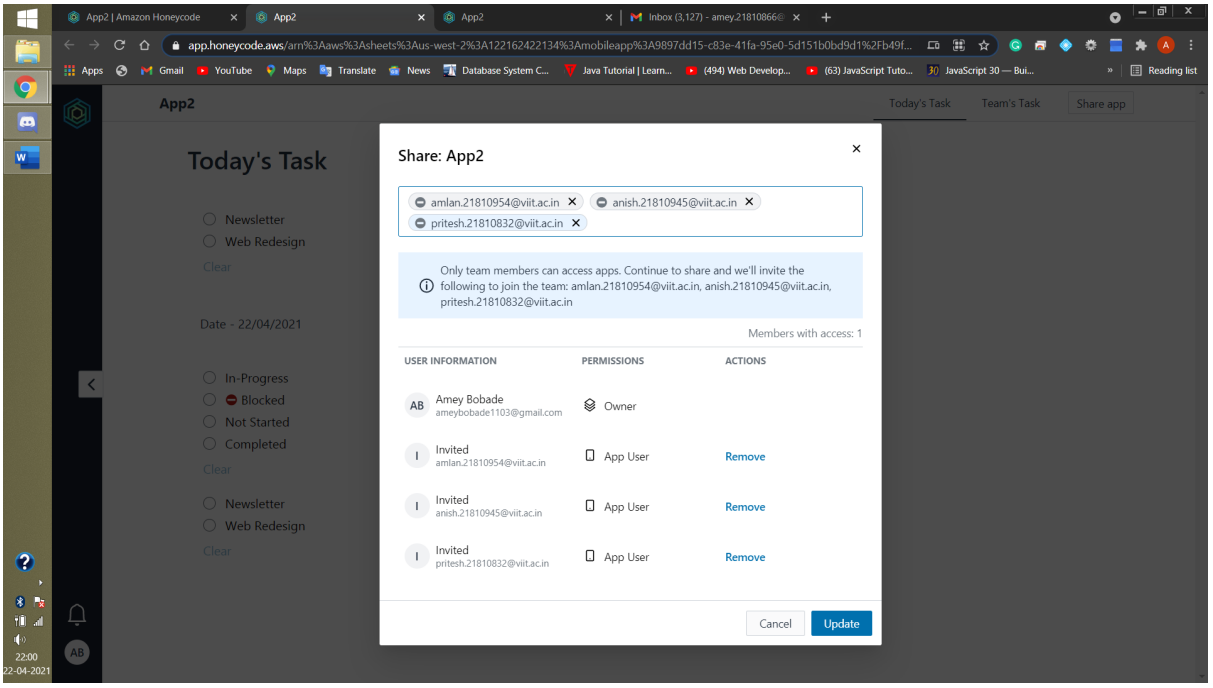
set it up.



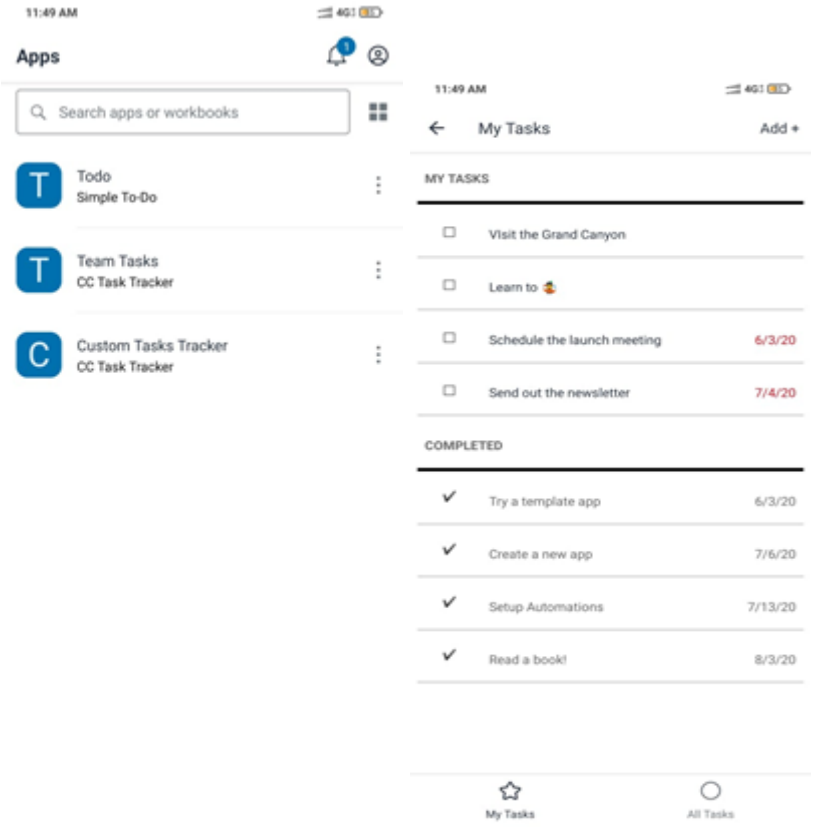
8) Go back to the Builder tab and make sure your app looks good, by checking it in the Preview tab.



9)Click on Share App, and add all your teammates' mails.



10)Install Amazon Honeycode app on your mobile phone for easy access. Lets see its various screens now.



And, with that we're done. A new custom app within 30 mins can be set up for your team!