

# RECOGNIZING DATA FROM IMAGES FOR HEALTH INSURANCE USING AZURE AI

## Contents

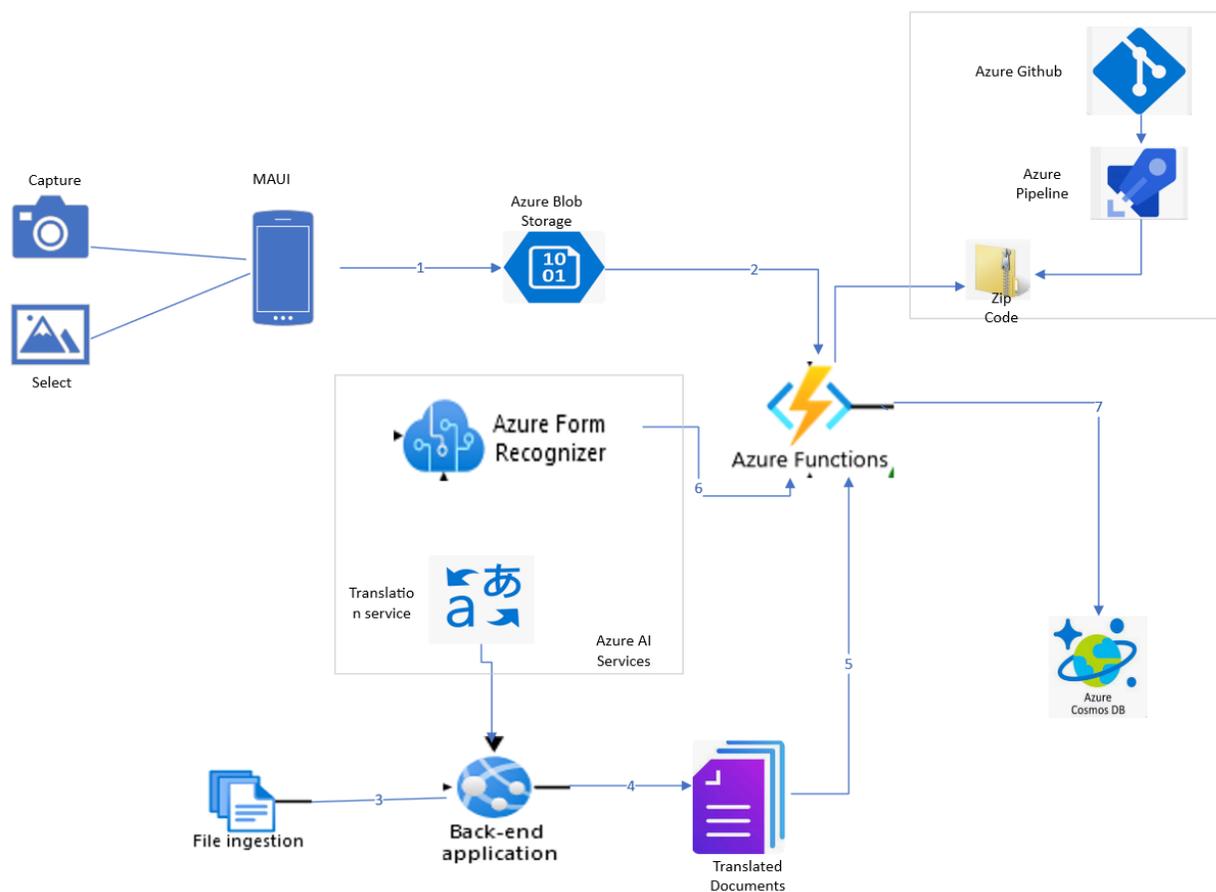
Problem Statement .....	3
Solution/Architecture .....	3
Technical Details and Implementation of solution.....	5
Mobile App (MAUI).....	5
Translation AI Service .....	6
Function App .....	9
Form Recognition AI Service .....	10
Azure Repository .....	11
Challenges in implementing the solution .....	12
Business Benefit .....	12

### Problem Statement

Healthcare and insurance have more relevance in today's world, especially when unexpected events like pandemic outbreaks and increased lifestyle health issues occur.

For insurance companies, it's a very complicated task to go through all the hospital bills and calculate the amount spent on hospitalization. It's an important task in the reimbursement of medical insurance.

### Solution/Architecture



In this solution,

- We are capturing the hospital bills using a **mobile app**, or we can select the existing image from Gallery.
- mobile app designed using the **MAUI framework**.

- Uploading these images into the **blob storage** account
- Meanwhile, we have the option of choosing the document in another language.
- using a **Translation AI service** to convert the document into the English language.
- Once the document is uploaded to the **storage account**, use the **function App** to extract the data from the images.
- Here, we are using a **form recognition AI service** to extract the data.
- This is the data we are inserting into the **Cosmos database**.
- We are maintaining the code base of the function app in the **Azure Git repository**.
- **Azure pipeline** used to build the solution
- Output of the function app is published into the **Azure resources**.

#### Data Extraction

Service dates	Type of service	Amount billed	Discount
ADVENTHEALTH ORLANDO, Claim # 7682223693581			
07/29/22	DRUGS	751.12	482.97
07/29/22	DRUGS	250.86	161.30
07/29/22	DRUGS	59.04	37.96
07/29/22	DRUGS	52.73	33.91
07/29/22	LABORATORY	298.00	191.61
07/29/22	LABORATORY	210.00	135.03
07/29/22	LABORATORY	74.00	47.58
07/29/22	LABORATORY	296.59	190.71
07/29/22	LABORATORY	971.18	624.47
07/29/22	OPERATING ROOM	11,603.20	7,460.86
07/29/22	ANESTHESIA SUP.	3,383.00	2,175.27
07/29/22	DIAG SERVICE	2,445.00	1,572.13
07/29/22	DRUGS	92.49	59.47
07/29/22	DRUGS	72.23	46.44
07/29/22	DRUGS	55.30	35.56
07/29/22	DRUGS	293.13	188.48
07/29/22	DRUGS	62.79	40.37
07/29/22	DRUGS	77.44	49.79
07/29/22	DRUGS	34.66	22.29
07/29/22	DRUGS	185.34	119.17
07/29/22	DRUGS	75.18	48.34
07/29/22	RECOVERY ROOM	2,892.93	1,860.17
<b>Total</b>		<b>\$24,236.21</b>	<b>\$15,583.88</b>

```
Document 0:
Item:
  Description: 'DRUGS', with confidence 0.97
  Amount: '751.12', with confidence 0.969
Item:
  Description: 'DRUGS', with confidence 0.97
  Amount: '250.86', with confidence 0.969
Item:
  Description: 'DRUGS', with confidence 0.97
  Amount: '59.04', with confidence 0.969
Item:
  Description: 'DRUGS', with confidence 0.97
  Amount: '52.73', with confidence 0.969
Item:
  Description: 'LABORATORY', with confidence 0.97
  Amount: '298', with confidence 0.968
Item:
  Description: 'LABORATORY', with confidence 0.97
  Amount: '210', with confidence 0.969
Item:
  Description: 'LABORATORY', with confidence 0.97
  Amount: '74', with confidence 0.969
Item:
  Description: 'LABORATORY', with confidence 0.97
  Amount: '296.59', with confidence 0.968
Item:
  Description: 'LABORATORY', with confidence 0.97
  Amount: '971.18', with confidence 0.969
Item:
  Description: 'OPERATING ROOM', with confidence 0.94
  Amount: '11003.2', with confidence 0.969
Item:
  Description: 'ANESTHESIA SUP.', with confidence 0.941
  Amount: '3383', with confidence 0.968
Item:
  Description: 'DIAG SERVICE', with confidence 0.94
  Amount: '2445', with confidence 0.969
Item:
  Description: 'DRUGS', with confidence 0.97
  Amount: '92.49', with confidence 0.969
Item:
  Description: 'DRUGS', with confidence 0.97
  Amount: '72.23', with confidence 0.969
Item:
  Description: 'DRUGS', with confidence 0.97
  Amount: '55.3', with confidence 0.969
Item:
  Description: 'DRUGS', with confidence 0.97
  Amount: '293.13', with confidence 0.967
Item:
```

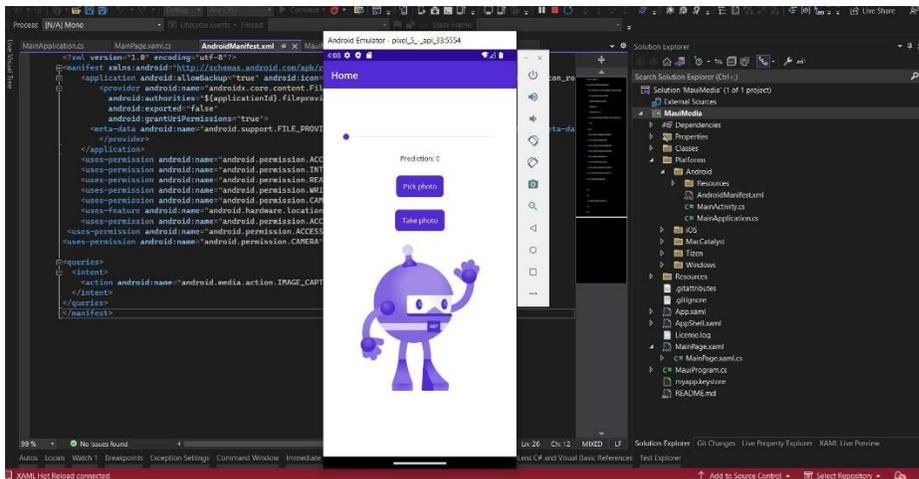
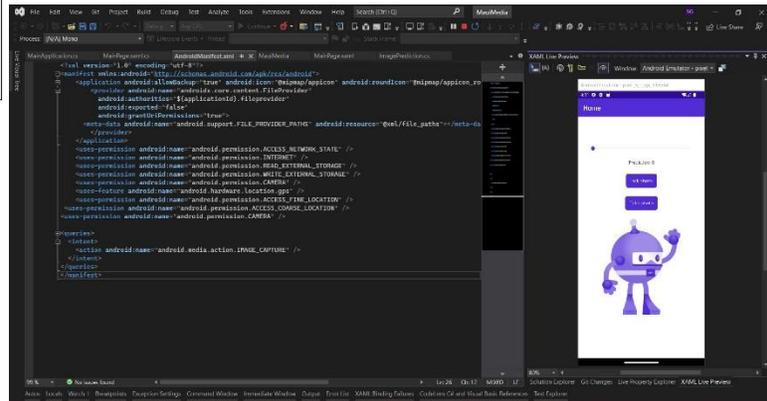
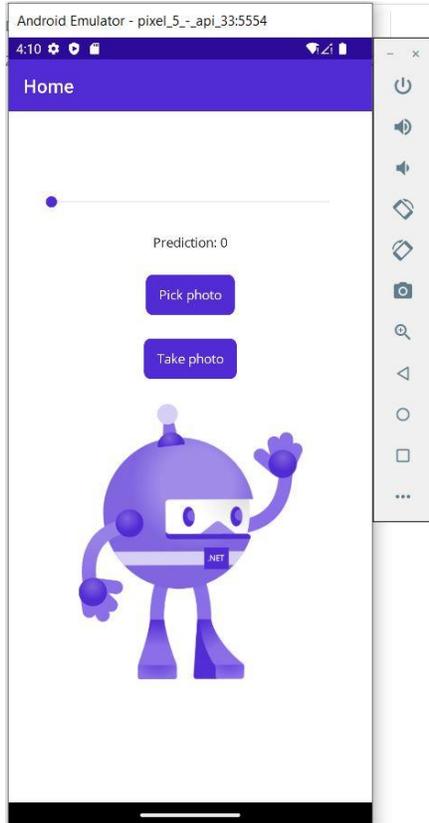
## Technical Details and Implementation of solution

### Mobile App (MAUI)

.NET Multi-platform App UI (.NET MAUI) is a cross-platform framework for creating native mobile and desktop apps with C# and XAML. Using .NET MAUI, you can develop apps that can run on Android, iOS, macOS, and Windows from a single shared code-base.

In this app, we are capturing and uploading the image into the storage account.

## Recognizing Data



## Translation AI Service

This service will get the document and convert into the English without changing the format and it will upload this document into the blob storage.

## Code:

```
static readonly string json = (" +
```

```
"{"inputs\": \" +
```

```

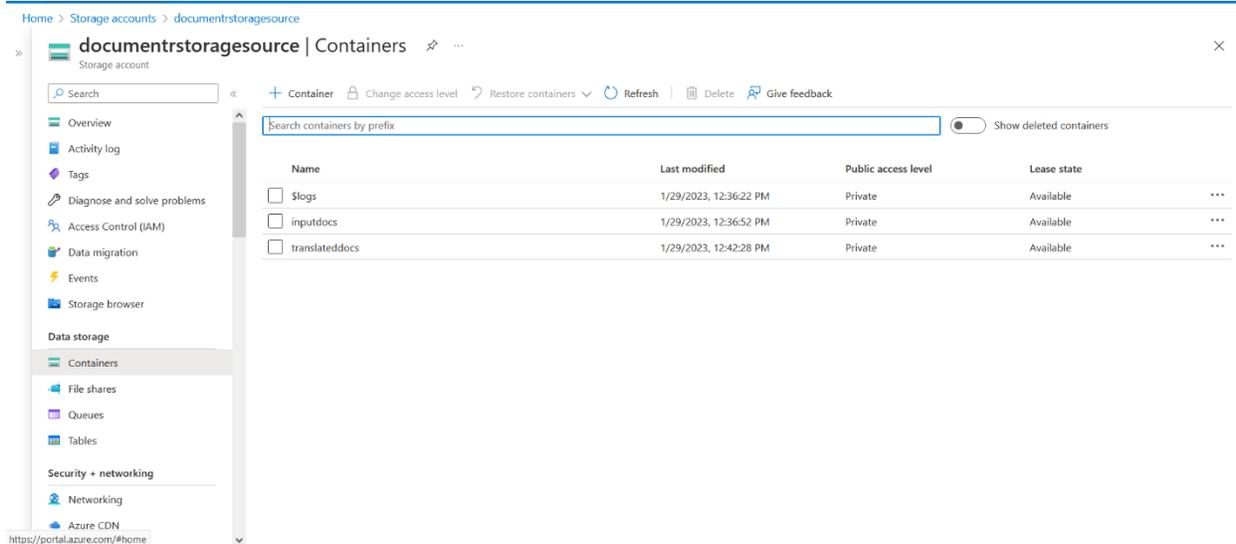
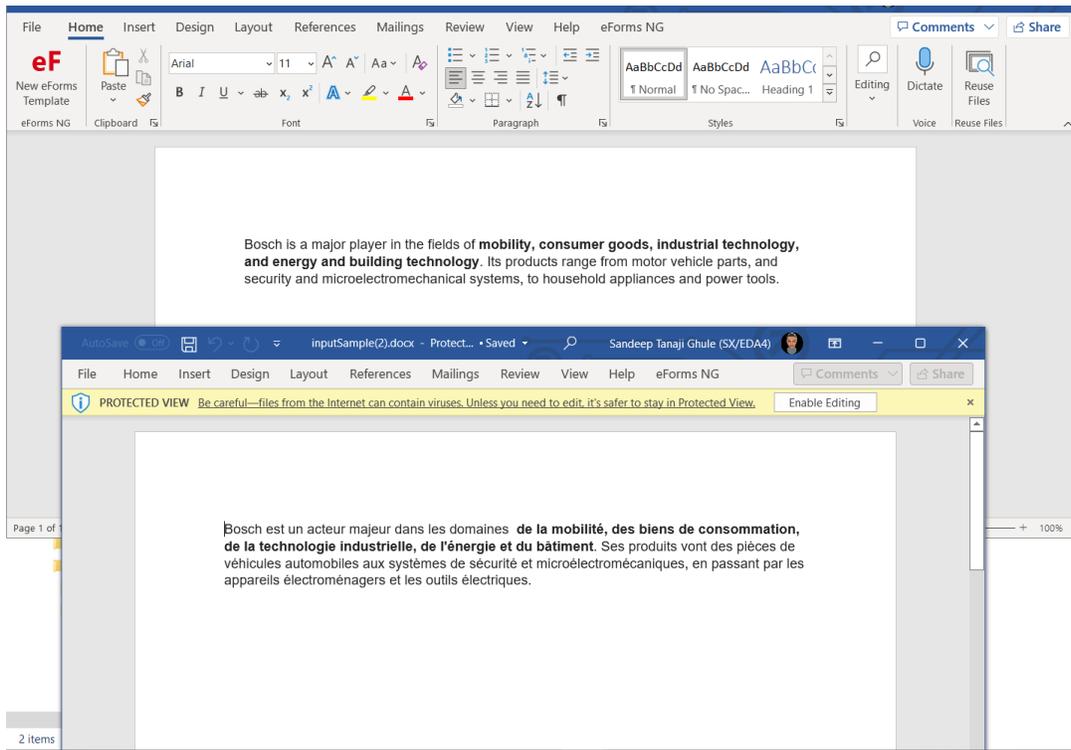
    [{"source\": " +
      {"sourceUrl\": \"<Sourcecontainer>\" +
        \"storageSource\": \"AzureBlob\"\" +
      },\" +
    \"targets\": \" +
      [{"targetUrl\": \"<TargetContainer>\" +
        \"storageSource\": \"AzureBlob\",\" +
        \"language\": \"fr\"}}]");
static async Task Main(string[] args)
{
    HttpClient client = new HttpClient();
    HttpRequestMessage request = new HttpRequestMessage();
    {
        StringContent data = new StringContent(json, Encoding.UTF8, "application/json");

        request.Method = HttpMethod.Post;
        request.RequestUri = new Uri(endpoint + route);
        request.Headers.Add("Ocp-Apim-Subscription-Key", key);
        request.Content = data;

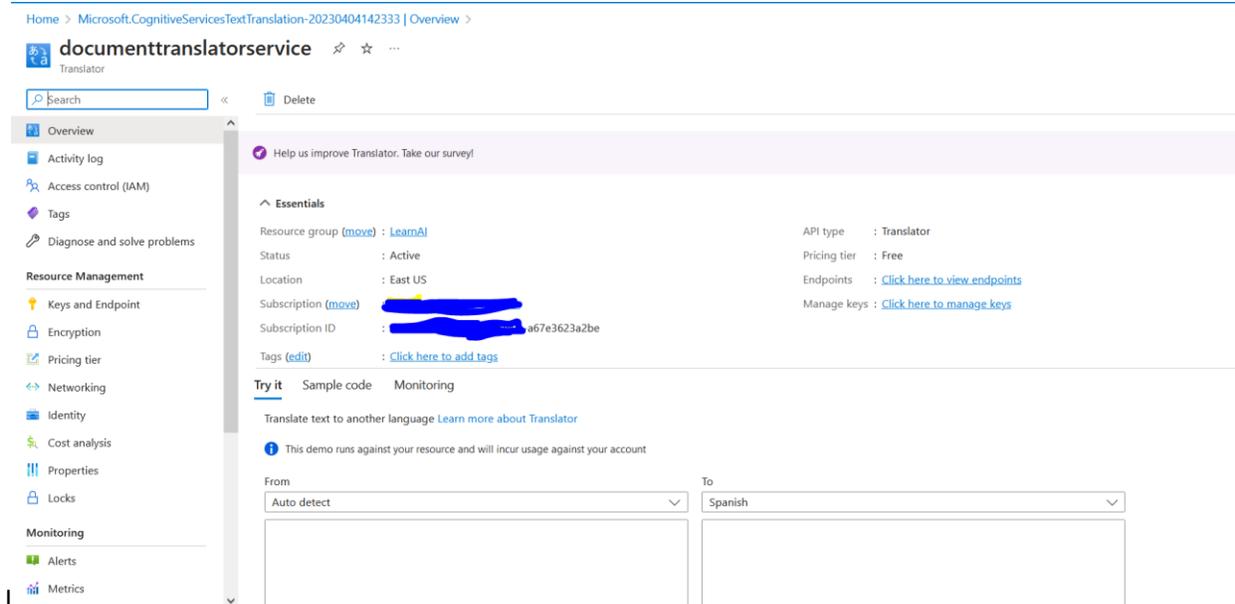
        HttpResponseMessage response = await client.SendAsync(request);
        string result = response.Content.ReadAsStringAsync().Result;
        if (response.IsSuccessStatusCode)
        {
            Console.WriteLine($"Operation successful with status code: {response.StatusCode}");
        }
        else
            Console.WriteLine($"Error occurred. Status code: {response.StatusCode}");
    }
}

```

## Recognizing Data



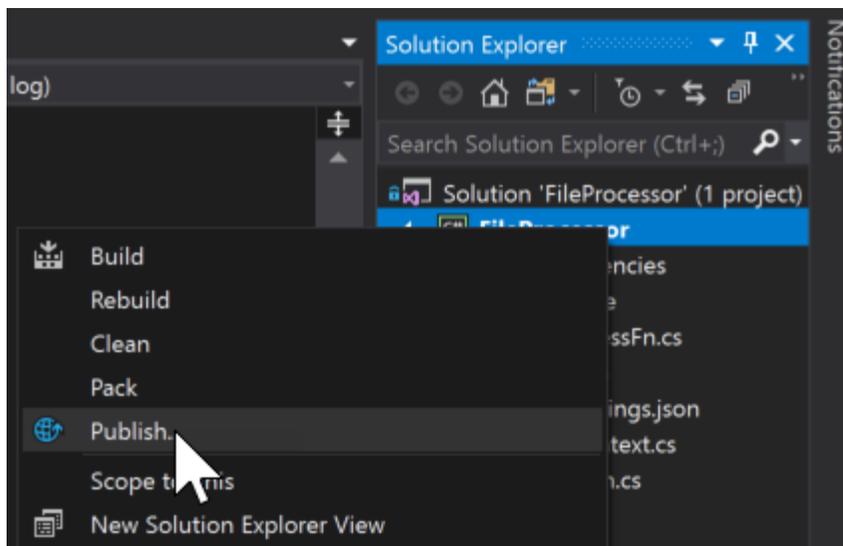
## Translator



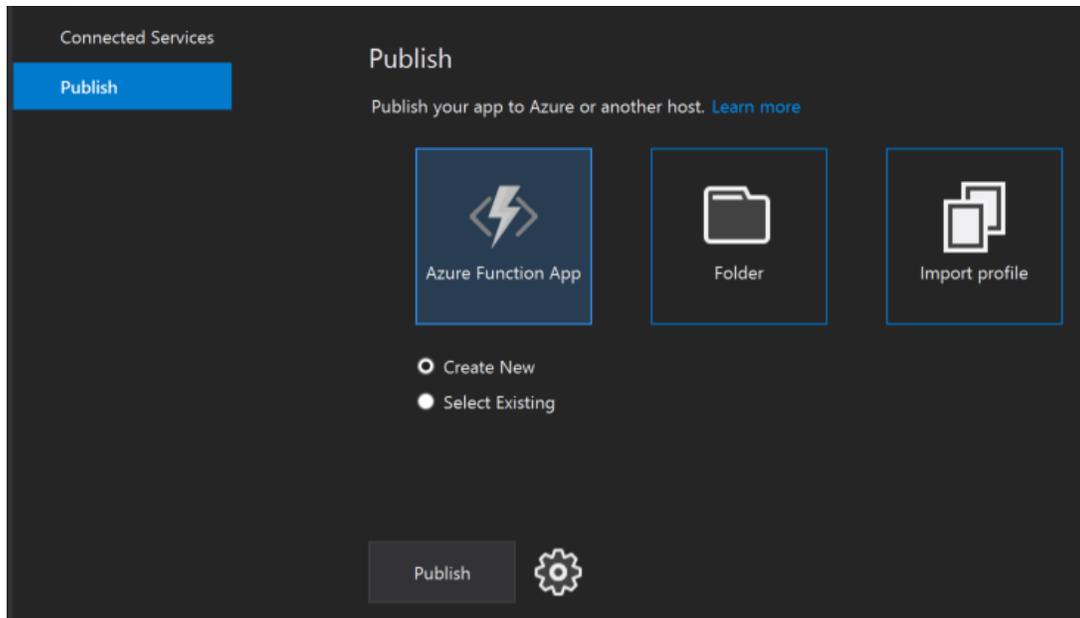
## Function App

The function app is responsible to get the file from storage account and extract the data from the image. Also this app is responsible to communicate with the Form recognition AI service.

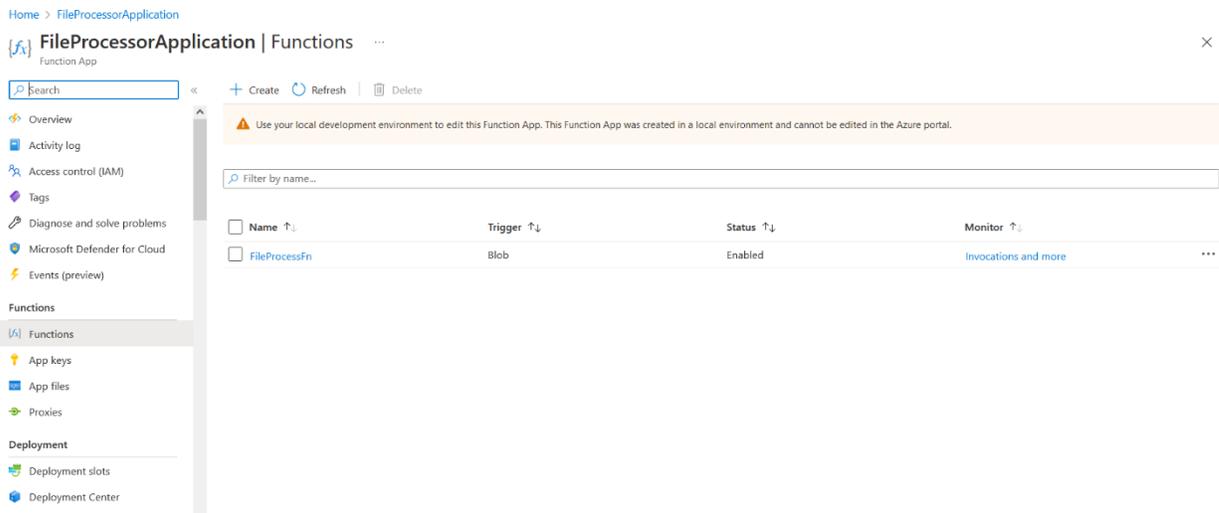
1. Inside Visual Studio, from the Solution Explorer right-click on the project name and choose Publish....



2. Choose Azure Function App, check Create New, and click Publish.



3. Give the app a unique name, choose your Subscription, and select the same Resource Group. For App Service Plan click New. . . .
4. Give the plan a unique name, choose the Location, and pick Consumption for the Size. Click OK.
5. Back in the Create App Service dialog, click Create.



### Form Recognition AI Service

This service is responsible to extract the data set from the image.

The screenshot shows the Azure portal interface for a Form Recognizer resource. The left sidebar contains navigation options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Resource Management, Keys and Endpoint, Encryption, Pricing tier, Networking, Identity, Cost analysis, Properties, Locks, Monitoring, Alerts, and Metrics. The main content area displays the resource details for 'formrecognizerininsurance'.

**Essentials**

- Resource group (move): [Learn AI](#)
- Status: Active
- Location: East US
- Subscription (move): [Redacted]
- Subscription ID: [Redacted]
- Tags (edit): [Click here to add tags](#)

**API type**: Form Recognizer  
**Pricing tier**: Free  
**Endpoint**: <https://formrecognizerininsurance.cognitiveservices.azure.com/>  
**Manage keys**: [Click here to manage keys](#)

**Get Started** | Monitoring

Learn more about what's new in the [latest Form Recognizer release](#)

**Form Recognizer Studio**  
 Extract text, key-value pairs, tables, and structures from documents automatically and accurately. Start with prebuilt models or create custom

**Client SDK and REST API**  
 Use the client SDK with the programming language of your choice or the REST API to automate the data extraction from your documents. Try it by following the links below.

The screenshot shows the Microsoft Visual Studio Debug Console with the following output:

```

Amount: '92.49', with confidence 0.969
Item:
  Description: 'DRUGS', with confidence 0.97
  Amount: '72.23', with confidence 0.969
Item:
  Description: 'DRUGS', with confidence 0.97
  Amount: '55.3', with confidence 0.969
Item:
  Description: 'DRUGS', with confidence 0.97
  Amount: '293.13', with confidence 0.967
Item:
  Description: 'DRUGS', with confidence 0.97
  Amount: '62.79', with confidence 0.969
Item:
  Description: 'DRUGS', with confidence 0.97
  Amount: '77.44', with confidence 0.967
Item:
  Description: 'DRUGS', with confidence 0.97
  Amount: '34.66', with confidence 0.97
Item:
  Description: 'DRUGS', with confidence 0.97
  Amount: '185.34', with confidence 0.969
Item:
  Description: 'DRUGS', with confidence 0.97
  Amount: '75.18', with confidence 0.969
Item:
  Description: 'RECOVERY ROOM', with confidence 0.94
  Amount: '2892.93', with confidence 0.969
Invoice Total: '$24236.21', with confidence 0.573

C:\Imp\Blog\formrecognizer\formrecognizer\bin\Debug\net6.0\formrecognizer.exe (process 41768) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
  
```

## Recognizing Data

The screenshot displays the Azure DevOps Pipelines interface. On the left, a sidebar shows navigation options like Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, and Deployment groups. The main area shows a job run for 'VSBuild' with a green checkmark and a duration of 20s. Below the job list, a console log for 'VSBuild' is visible, showing the start of the process and a list of loaded DLLs, including System.Security.Cryptography, System.Net.Security, System.Collections, NonGeneric, System.Memory, System.Linq.Expressions, System.Text.Json, System.Text.Encoding.Web, System.Memory.Data, System.Runtime.Intrinsics, System.Numerics.Vectors, System.Net.Sockets, System.Net.NameResolution, System.Threading.ThreadPool, System.Security.Cryptography, System.Runtime.CompilerServices, System.Text.Encoding.Extensions, System.Threading.Thread, System.Drawing.Primitives, and System.Xml.ReaderWriter. The log ends with 'The program "[41768] fornrecognizer.exe" has exited with code 0 (0x0)'.

### GitHub Links

#### Translator AI service

<https://github.com/KomalHonmane/AzureBlogathon/tree/main/DocumentTranslator>

#### Form Recognition console app

<https://github.com/KomalHonmane/AzureBlogathon/tree/main/fornrecognizer>

#### Function App- Form Recognition

[https://github.com/KomalHonmane/AzureBlogathon/tree/main/FunctionApp\\_FormRecognizer](https://github.com/KomalHonmane/AzureBlogathon/tree/main/FunctionApp_FormRecognizer)

#### MAUI – Mobile App

[https://github.com/KomalHonmane/AzureBlogathon/tree/main/MobileApp\\_MAUI](https://github.com/KomalHonmane/AzureBlogathon/tree/main/MobileApp_MAUI)

### Challenges in implementing the solution

- Creating mobile app with this new framework MAUI
- To implement the form recognition service
- Upload document in storage account

### Business Benefit

Automate data extraction from various hospital bill formats so that healthcare and insurance companies can easily calculate the amount spent in hospitals and offer better insurance to their customers.

Additional advantages include automated claim processing with reduced human efforts to benefit both payors and insurance providers.