Personal Portfolio

Presentations are communication tools.

About Me

David Nababan

I am a student of Electrical Engineering with proficiency in backend development using node.js. I also have expertise in IT infrastructure and working with the Google Cloud Platform.


linkedin.com/in/davidnababanee/


nbbndvd@gmail.com


https://github.com/DavidNbEE

Let's Connect

Educational Background

Bangkit Academy

2023

Cloud Computing

Utilize JavaScript, node.Js, Google Cloud Platform

Sepuluh Nopember Institute Of Technology

2020-present

Electrical Engineering

Skills

Here are my skills!

Google Cloud Platform

DigiNote

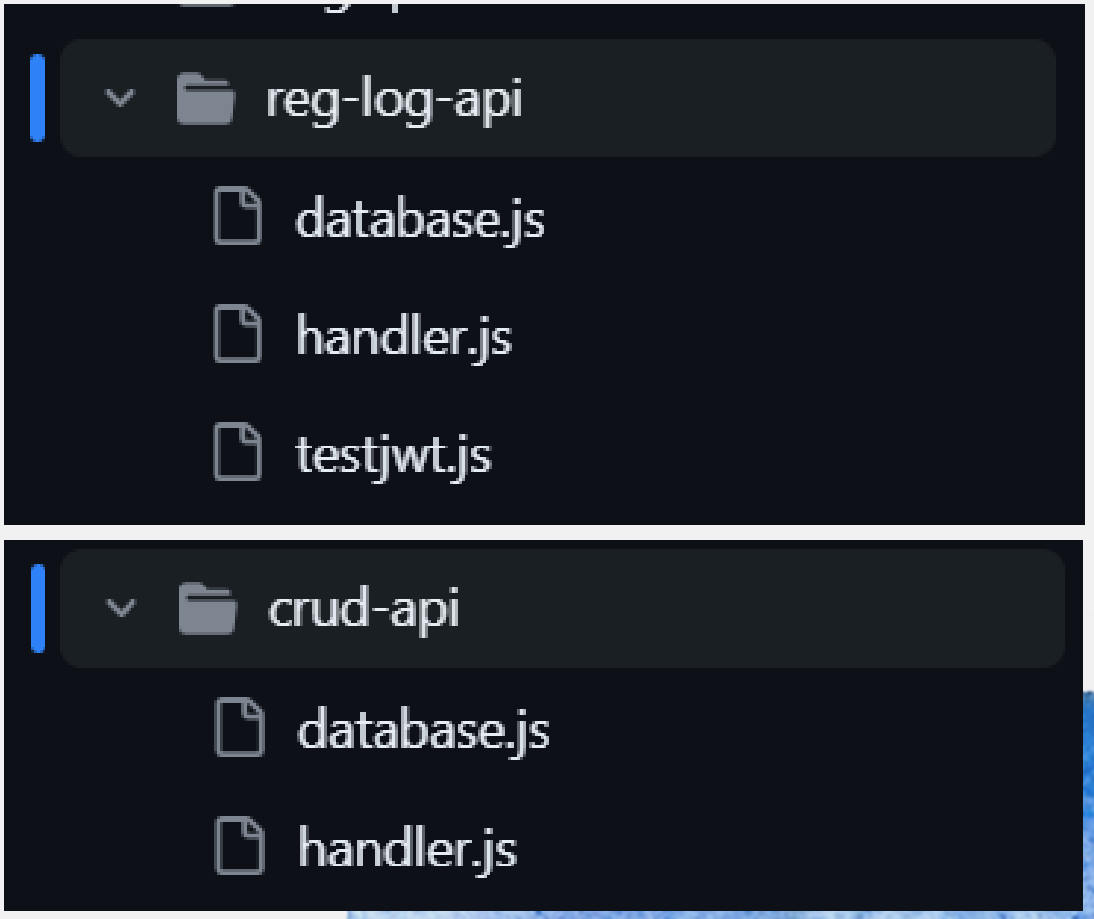
DigiNote is a Note taking app with feature of OCR that extract text from object that taken. Scan your handwritten note and make it to digital!

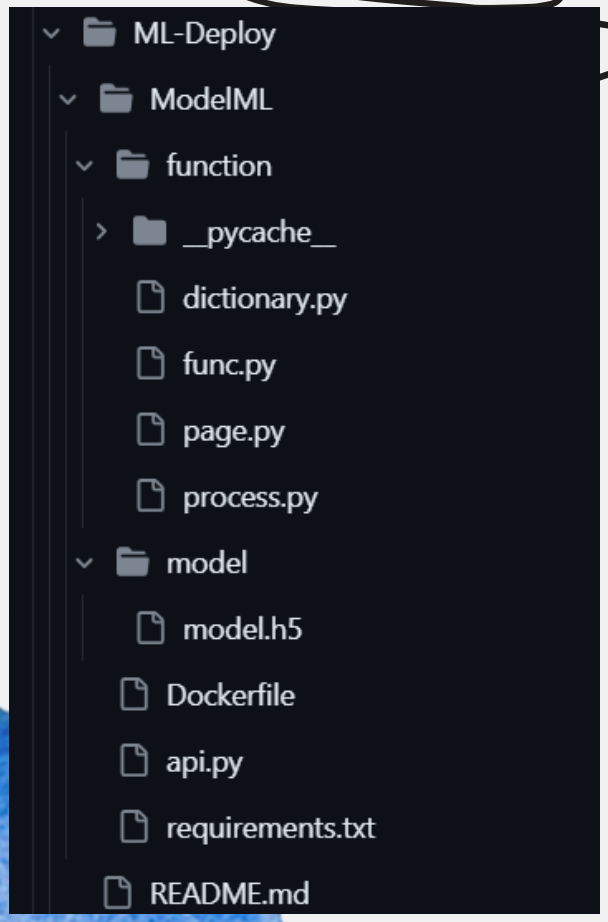
What I Do?

We are working with 5 person to create this app with different skills like Machine Learning, Cloud Computing, and Mobile Development.
As A Cloud Computing, I did ... :

1. **Created Backend with restAPI using Node.Js with Express framework. I created API for user register and login also for the notes.**
2. **Created API for Machine Learning Model and deploy it with Cloud Run**
3. **Deployed the Application with Google Cloud Platform**
4. **Created and managed database with SQL in CloudSQL for users and note data**

RestAPI


reg-log-api
database.js
handler.js
testjwt.js
crud-api
database.js
handler.js


ML-Deploy
ModelML
function
__pycache__
dictionary.py
func.py
page.py
process.py
model
model.h5
Dockerfile
api.py
requirements.txt
README.md

RestAPI

Code Blame 10 lines (8 loc) · 211 Bytes

```
1   const mysql =  require('mysql')
2
3   const pool = mysql.createPool({
4       host: "your_host",
5       user: "your_user",
6       password: "your_password",
7       database: "your_database",
8     })
9
10  module.exports = pool
```

code for connecting to database

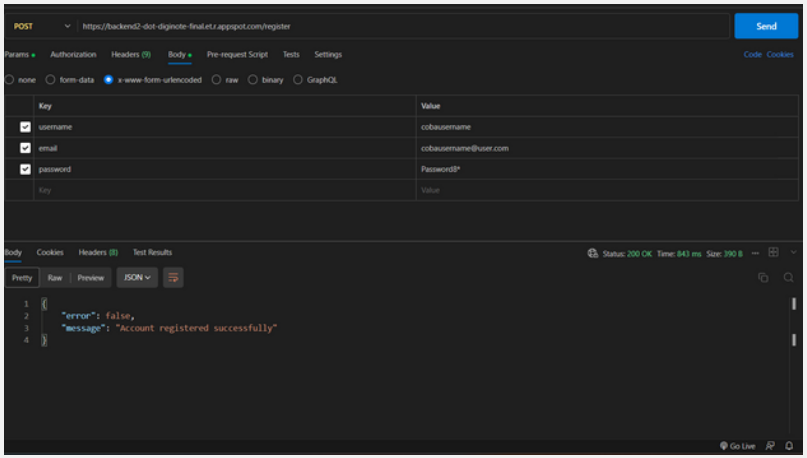
```
8   const registerHandler = (req, res) => {
9       const { username, email, password } = req.body;
10
11      if (!username || !email || !password) {
12        return res.status(400).json({ error: true, message: 'All fields are required' });
13      }
14
15      // Ketentuan Passwordnya
16      if (password.length < 8) {
17        return res.status(400).json({ error: true, message: 'Password must be at least 8 characters long' });
18      }
19      if (!/[A-Z]/.test(password)) {
20        return res.status(400).json({ error: true, message: 'Password must contain at least one uppercase letter' });
21      }
22      if (!/[0-9]/.test(password)) {
23        return res.status(400).json({ error: true, message: 'Password must contain at least one number' });
24      }
25      if (!/[!@#$%^&.*_]/.test(password)) {
26        return res.status(400).json({ error: true, message: 'At least one special character (! @ # $ % ^ & * . _)' });
27      }
28
29      const checkEmailQuery = 'SELECT * FROM users WHERE email = ?';
30      pool.query(checkEmailQuery, [email], (error, emailResults) => {
31        if (error) {
32          console.error('Error checking email:', error);
33          return res.status(500).json({ error: true, message:  'Server error' });
34        }
35
36        if (emailResults.length > 0) {
37          return res.status(409).json({ error: true, message: 'Email is already taken' });
38        }
```

Code example for login and register handler

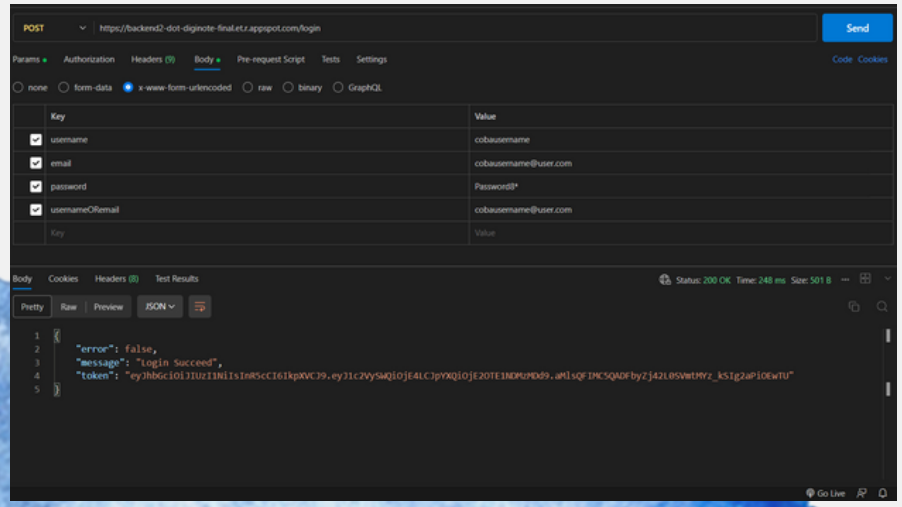
```
1   const express = require('express')
2   const Multer = require('multer')
3   const { registerHandler, loginHandler } = require('./reg-log-api/handler')
4   const { getAllNoteHandler, createNoteHandler, getNoteIdHandler, editNoteHandler, deleteNoteHa
5   const router = express.Router()
6   const imgUpload = require ('./imgUploads/imgUpload')
7
8   const multer = Multer({
9       storage: Multer.MemoryStorage,
10      fileSize: 5 * 1024 * 1024
11  })
12
13  router.post('/register', registerHandler)
14
15  router.post('/login', loginHandler)
16
17  router.post('/notes',multer.single('image'), imgUpload.uploadToGcs, createNoteHandler)
18
19  router.get('/notes', getAllNoteHandler)
20
21  router.get('/notes/:noteId', getNoteIdHandler)
22
23  router.post('/notes/edit/:noteId', editNoteHandler)
24
25  router.delete('/notes/delete/:noteId', deleteNoteHandler)
26
27  router.post('/upload', multer.single('image'), imgUpload.uploadToGcs, uploadimghandler )
28
29  module.exports = router
```

In making of this APP API, I tend to differentiate file to easy manage such as server file, routes file, handler file (to create the program logic/algorithm)

Postman Test

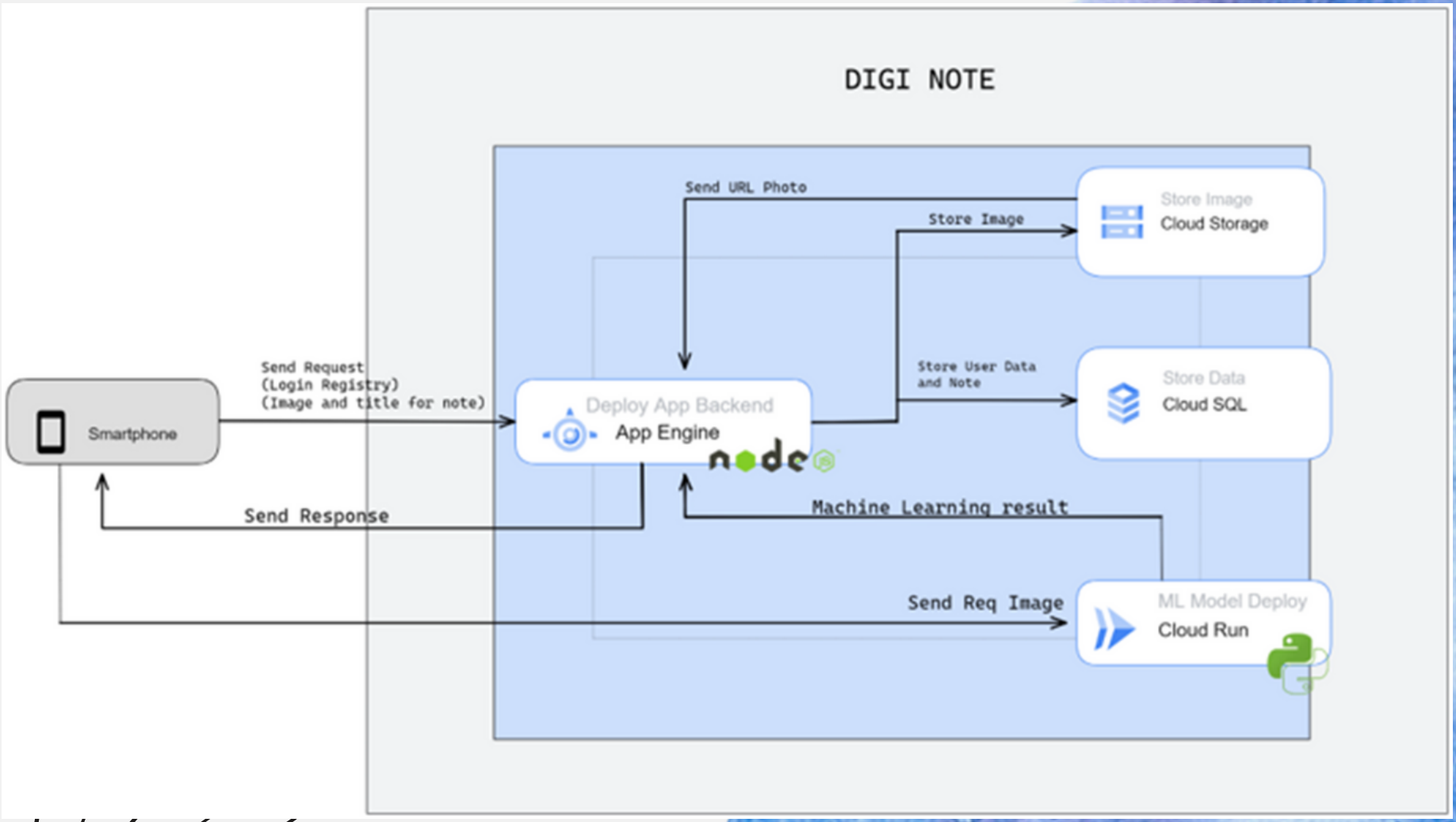


Login Test



Register Test

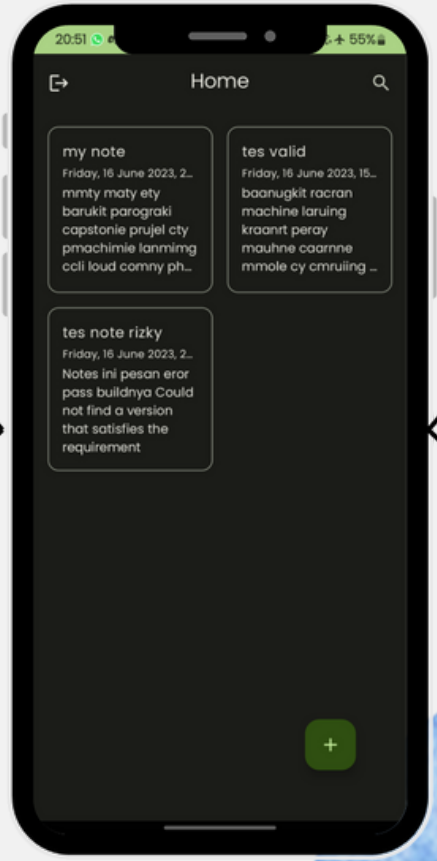
Deployment


DIGI NOTE
Send URL Photo
Store Image
Store Image
Cloud Storage
Send Request
(Login Registry)
(Image and title for note)
Store User Data
and Note
Store Data
Cloud SQL
Smartphone
Deploy App Backend
App Engine
node
Send Response
Machine Learning result
Send Req Image
ML Model Deploy
Cloud Run

Application Overview

NOTE DITING

LOGIN/ SIGNUP



VIEW ORIGINAL NOTE

NOTE CAPTURE

Full Documentation

FOR FULL DOCUMENTATION OF THIS PROJECT CAN BE ACESSED AT
GITHUB.COM/FFATHAN/DIGINOTE

Thank You