**Google Auth using Firebase**

**How to do Google Authentication (SSO) using Firebase**

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:google\_sign\_in/google\_sign\_in.dart';

import 'package:googleapis/drive/v3.dart' as drive;

import 'package:googleapis/sheets/v4.dart' as sheets;

import 'package:priti\_app/data/card\_data.dart';

import 'package:priti\_app/services/google\_auth.dart';

GoogleSignIn googleSignIn = GoogleSignIn(

scopes: [

'email',

drive.DriveApi.driveScope,

],

);

Map<String, String> authHeaders;

GoogleAuthClient authenticateClient;

drive.DriveApi driveApi;

sheets.SheetsApi sheetApi;

GoogleSignInAuthentication googleAuth;

class Auth {

Future<User> googleLogin() async {

try {

final GoogleSignInAccount account = await googleSignIn.signIn();

authHeaders = await account.authHeaders;

authenticateClient = GoogleAuthClient(authHeaders);

driveApi = drive.DriveApi(authenticateClient);

sheetApi = sheets.SheetsApi(authenticateClient);

googleAuth = await account.authentication;

final GoogleAuthCredential credential = GoogleAuthProvider.credential(

accessToken: googleAuth.accessToken,

idToken: googleAuth.idToken,

);

final UserCredential response =

await FirebaseAuth.instance.signInWithCredential(credential);

assert(response.user.email != null);

assert(response.user.displayName != null);

assert(!response.user.isAnonymous);

assert(await response.user.getIdToken() != null);

final User currentUser = FirebaseAuth.instance.currentUser;

assert(response.user.uid == currentUser.uid);

return response.user;

} catch (e) {

print(e);

return null;

}

}

Future<sheets.Spreadsheet> createSheet() async {

sheets.SpreadsheetProperties spreadsheetProperties =

sheets.SpreadsheetProperties();

spreadsheetProperties.title = "Priti Contacts";

sheets.SheetProperties sheetProperties = sheets.SheetProperties();

sheetProperties.title = "Sheet1";

sheets.Spreadsheet spreadsheet = sheets.Spreadsheet();

spreadsheet.properties = spreadsheetProperties;

sheets.Sheet sheet = sheets.Sheet();

sheet.data = List<sheets.GridData>();

sheets.GridData gg = sheets.GridData();

gg.rowData = List<sheets.RowData>();

sheets.RowData dss = sheets.RowData();

dss.values = List<sheets.CellData>();

for (String item in columnNames) {

sheets.CellData cd = sheets.CellData();

cd.userEnteredValue = sheets.ExtendedValue();

cd.userEnteredValue.stringValue = item;

cd.textFormatRuns = List<sheets.TextFormatRun>();

sheets.TextFormatRun r = sheets.TextFormatRun();

r.format = sheets.TextFormat();

r.format.bold = true;

r.format.fontSize = 14;

cd.textFormatRuns.add(r);

dss.values.add(cd);

}

gg.rowData.add(dss);

sheet.data.add(gg);

sheet.properties = sheetProperties;

spreadsheet.sheets = List<sheets.Sheet>();

spreadsheet.sheets.add((sheet));

print(spreadsheet.sheets);

sheets.Spreadsheet create = await sheetApi.spreadsheets.create(spreadsheet);

return create;

}

static List<String> columnNames = [

'Id',

"Name",

"Phone",

"Phone2",

"Email",

"Email2",

"Address",

"Website",

"Zip Code",

"Company",

"Job Title",

"DateTime",

"Notes"

];

}