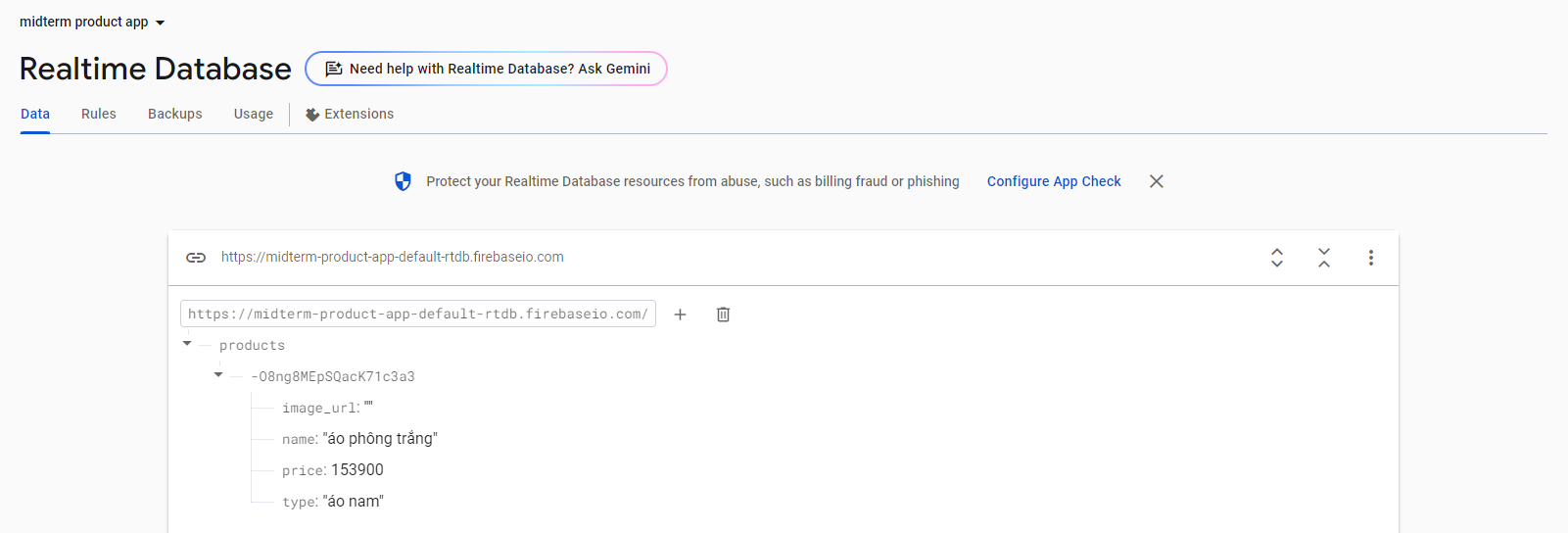
**Họ và tên: Đoàn Trọng Tấn**

**Mã sinh viên: 21IT106**

**CÁC BƯỚC THỰC HIỆN ĐỀ TÀI**

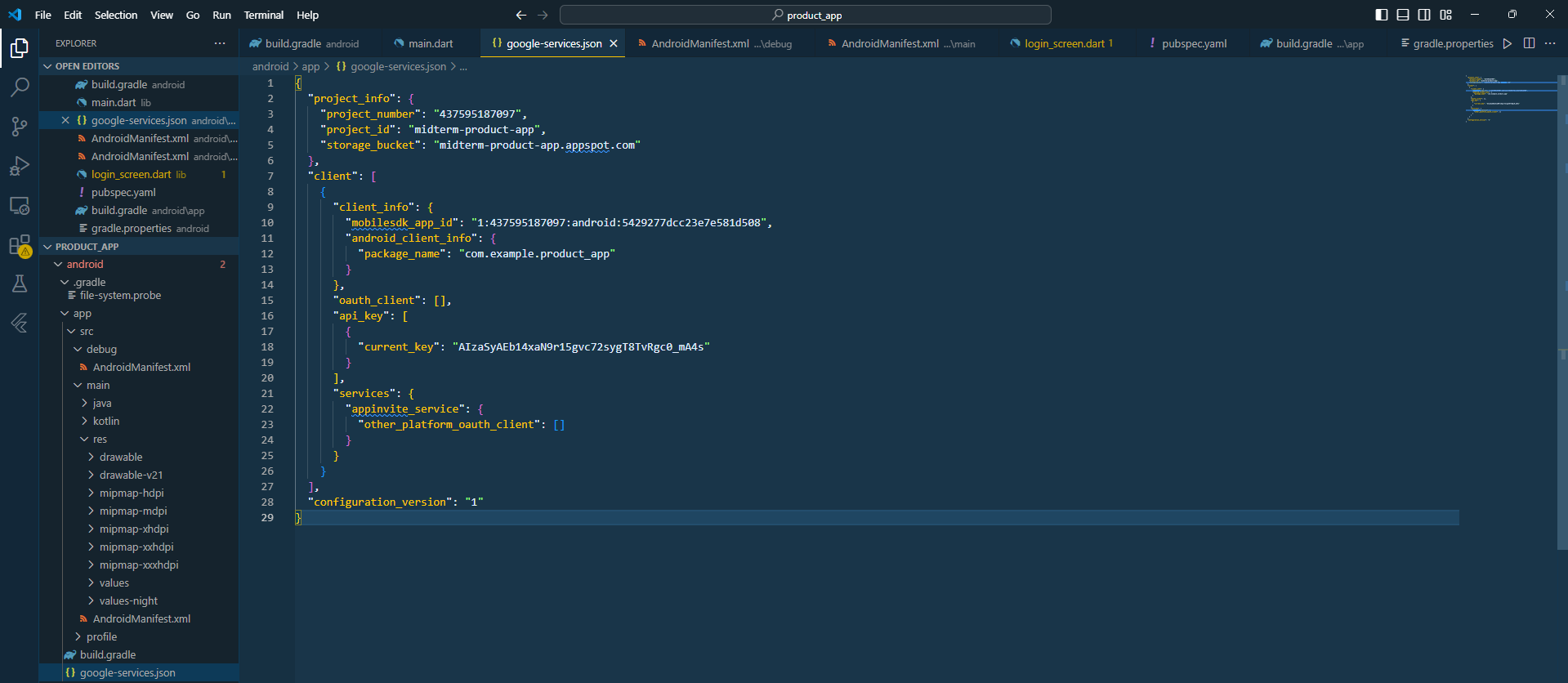
**Bước 1:** Vào Firebase FireStore Database tạo database midterm product app sau dó nhấn Get start Realtime Database, thiết lập Rule.



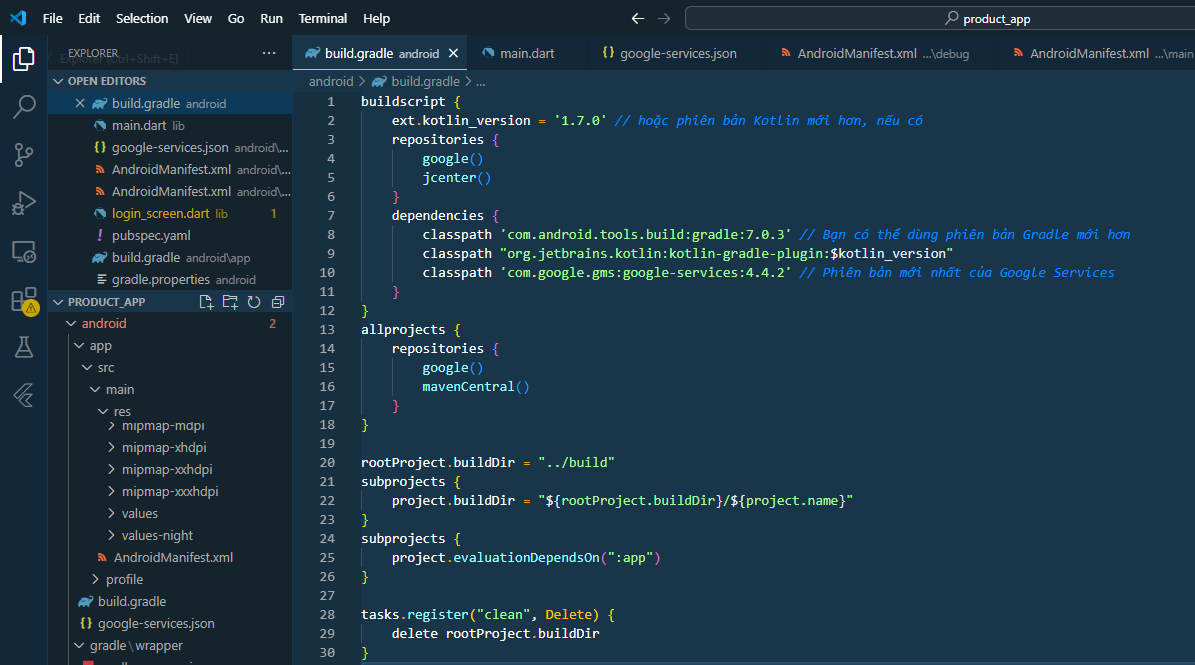
**Bước 2:** Tạo dự án Flutter có tên Product app

**Bước 3:** Cấu hình Firebase cho dự án

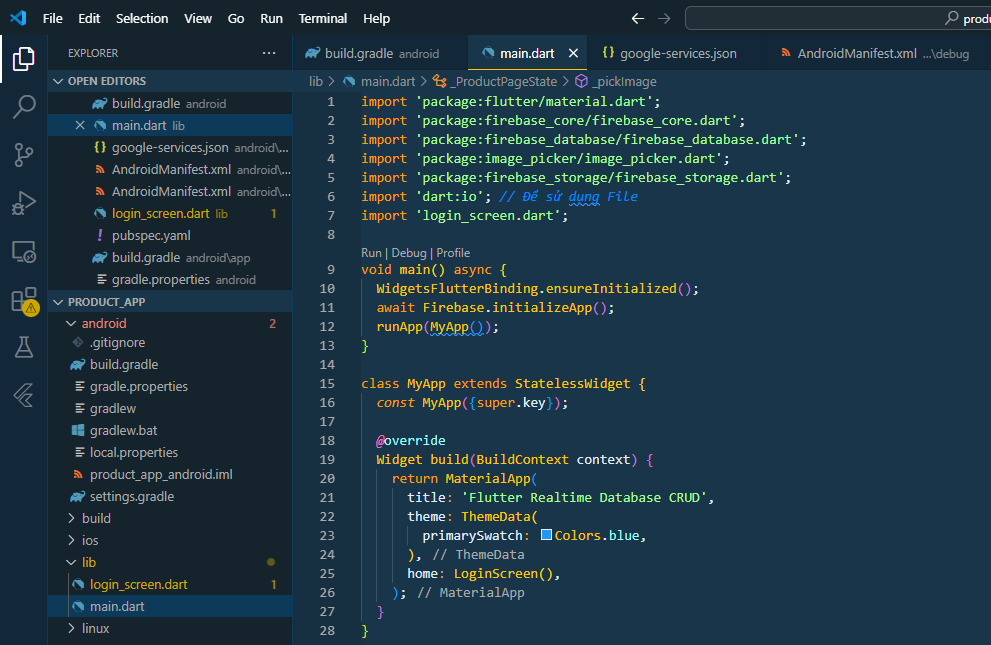
* Tải tệp cấu hình google-services.json và đặt vào thư mục android/app.



* Cập nhật tệp android/build.gradle



* Khởi tạo firebase trong ứng dụng



Bước 4: Xây dựng chức năng thêm sản phẩm

* **Giao diện người dùng để nhập thông tin sản phẩm**: Tạo các trường nhập liệu cho tên, loại, giá sản phẩm và nút chọn hình ảnh:

class \_ProductPageState extends State<ProductPage> {

*final* TextEditingController nameController = TextEditingController();

*final* TextEditingController typeController = TextEditingController();

*final* TextEditingController priceController = TextEditingController();

  File? \_image;

*// Hàm chọn hình ảnh*

  Future<void> \_pickImage() async {

*final* pickedFile = await ImagePicker().pickImage(source: ImageSource.gallery);

    setState(() {

      if (pickedFile != null) {

        \_image = File(pickedFile.path);

      } else {

        print('Không có hình ảnh nào được chọn.');

      }

    });

  }

*// Hàm tải lên hình ảnh và lấy URL*

  Future<String?> \_uploadImage(File image) async {

    try {

      String fileName = DateTime.now().millisecondsSinceEpoch.toString();

      Reference ref = FirebaseStorage.instance.ref().child('product\_images/$fileName');

      UploadTask uploadTask = ref.putFile(image);

*// Kiểm tra trạng thái upload*

      uploadTask.snapshotEvents.listen((TaskSnapshot snapshot) {

      print('Progress: ${(snapshot.bytesTransferred / snapshot.totalBytes) \* 100} %');

      });

      TaskSnapshot snapshot = await uploadTask;

      String downloadUrl = await snapshot.ref.getDownloadURL();

      print('Download URL: $downloadUrl'); *// In ra URL tải về*

      return downloadUrl;

    } catch (e) {

      print('Lỗi khi tải lên hình ảnh: $e');

      return null;

    }

  }

*@*override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: *const* Text('Dữ liệu sản phẩm'),

      ),

      body: Padding(

        padding: *const* EdgeInsets.all(8.0),

        child: Column(

          children: [

*// Các trường nhập liệu*

            TextField(

              controller: nameController,

              decoration: *const* InputDecoration(labelText: 'Tên sản phẩm'),

            ),

            TextField(

              controller: typeController,

              decoration: *const* InputDecoration(labelText: 'Loại sản phẩm'),

            ),

            TextField(

              controller: priceController,

              decoration: *const* InputDecoration(labelText: 'Giá sản phẩm'),

              keyboardType: TextInputType.number,

            ),

*const* SizedBox(height: 10),

*// Hiển thị hình ảnh đã chọn*

            \_image == null

                ? Text('Chưa chọn hình ảnh')

                : Image.file(\_image!, height: 100),

            ElevatedButton(

              onPressed: \_pickImage,

              child: Text('Chọn hình ảnh'),

            ),

*const* SizedBox(height: 20),

            ElevatedButton(

              onPressed: () async {

                String? imageUrl;

                if (\_image != null) {

                  imageUrl = await \_uploadImage(\_image!);

                  print('Image URL: $imageUrl');

                } else {

                  imageUrl = '';

                }

*// Thêm sản phẩm vào Realtime Database*

                DatabaseReference ref = FirebaseDatabase.instance.ref("products").push();

                await ref.set({

                  'name': nameController.text,

                  'type': typeController.text,

                  'price': int.parse(priceController.text),

                  'image\_url': imageUrl,

                });

                nameController.clear();

                typeController.clear();

                priceController.clear();

                setState(() {

                  \_image = null;

                });

*// Hiển thị thông báo thành công*

                ScaffoldMessenger.of(context).showSnackBar(

                  SnackBar(content: Text('Thêm sản phẩm thành công')),

                );

              },

              child: *const* Text('Thêm sản phẩm'),

            ),

*const* SizedBox(height: 20),

            Expanded(

              child: StreamBuilder(

                stream: FirebaseDatabase.instance.ref("products").onValue,

                builder: (context, AsyncSnapshot<DatabaseEvent> snapshot) {

                  if (!snapshot.hasData || snapshot.data!.snapshot.value == null) {

                    return *const* Center(child: CircularProgressIndicator());

                  }

                  Map<*dynamic*, *dynamic*> products = snapshot.data!.snapshot.value as Map<*dynamic*, *dynamic*>;

                  List<Item> items = [];

                  products.forEach((key, value) {

                    items.add(Item(key: key, data: value));

                  });

                  return ListView.builder(

                    itemCount: items.length,

                    itemBuilder: (context, index) {

                      var item = items[index];

                      return ListTile(

                        title: Text(item.data['name']),

                        subtitle: Text('Giá: ${item.data['price']}'),

                        leading: item.data['image\_url'] != null && item.data['image\_url'] != ''

                            ? Image.network(item.data['image\_url'], width: 50, height: 50,

                            fit: BoxFit.cover, *// Đảm bảo hình ảnh vừa với ô*

                            errorBuilder: (context, error, stackTrace) {

*// Xử lý lỗi nếu không tải được hình ảnh*

                              return Icon(Icons.broken\_image);

                              },

                            )

                            : Icon(Icons.image), *// Biểu tượng mặc định nếu không có hình ảnh*

                        onTap: () {

                          Navigator.push(

                            context,

                            MaterialPageRoute(

                              builder: (context) => ProductDetailPage(product: item),

                            ),

                          );

                        },

                        trailing: Row(

                          mainAxisSize: MainAxisSize.min,

                          children: [

                            IconButton(

                              icon: Icon(Icons.edit),

                              onPressed: () {

                                \_editProduct(item);

                              },

                            ),

                            IconButton(

                              icon: Icon(Icons.delete),

                              onPressed: () {

                                FirebaseDatabase.instance.ref("products/${item.key}").remove();

                              },

                            ),

                          ],

                        ),

                      );

                    },

                  );

                },

              ),

            ),

          ],

        ),

      ),

    );

  }

**Bước 6: Hiển thị danh sách sản phẩm**

* Sử dụng StreamBuilder để hiển thị danh sách các sản phẩm được lưu trong Firebase Realtime Database:
* Expanded(
* child: StreamBuilder(
* stream: FirebaseDatabase.instance.ref("products").onValue,
* builder: (context, AsyncSnapshot<DatabaseEvent> snapshot) {
* if (!snapshot.hasData || snapshot.data!.snapshot.value == null) {
* return *const* Center(child: CircularProgressIndicator());
* }
* Map<*dynamic*, *dynamic*> products = snapshot.data!.snapshot.value as Map<*dynamic*, *dynamic*>;
* List<Item> items = [];
* products.forEach((key, value) {
* items.add(Item(key: key, data: value));
* });
* return ListView.builder(
* itemCount: items.length,
* itemBuilder: (context, index) {
* var item = items[index];
* return ListTile(
* title: Text(item.data['name']),
* subtitle: Text('Giá: ${item.data['price']}'),
* leading: item.data['image\_url'] != null && item.data['image\_url'] != ''
* ? Image.network(item.data['image\_url'], width: 50, height: 50,
* fit: BoxFit.cover, *// Đảm bảo hình ảnh vừa với ô*
* errorBuilder: (context, error, stackTrace) {
* *// Xử lý lỗi nếu không tải được hình ảnh*
* return Icon(Icons.broken\_image);
* },
* )
* : Icon(Icons.image), *// Biểu tượng mặc định nếu không có hình ảnh*
* onTap: () {
* Navigator.push(
* context,
* MaterialPageRoute(
* builder: (context) => ProductDetailPage(product: item),
* ),
* );
* },
* trailing: Row(
* mainAxisSize: MainAxisSize.min,
* children: [
* IconButton(
* icon: Icon(Icons.edit),
* onPressed: () {
* \_editProduct(item);
* },
* ),
* IconButton(
* icon: Icon(Icons.delete),
* onPressed: () {
* FirebaseDatabase.instance.ref("products/${item.key}").remove();
* },
* ),
* ],
* ),
* );
* },
* );
* },
* ),
* ),

**Bước 7: Chỉnh sửa và xóa sản phẩm**

* Thêm chức năng chỉnh sửa và xóa sản phẩm trong danh sách bằng cách sử dụng IconButton:
* IconButton(
* icon: Icon(Icons.edit),
* onPressed: () {
* \_editProduct(item);
* },
* ),
* IconButton(
* icon: Icon(Icons.delete),
* onPressed: () {
* FirebaseDatabase.instance.ref("products/${item.key}").remove();
* },
* ),
* Hàm chỉnh sửa sản phẩm
* void \_editProduct(Item item) {
* nameController.text = item.data['name'];
* typeController.text = item.data['type'];
* priceController.text = item.data['price'].toString();
* \_image = null; *// Đặt lại hình ảnh*
* String existingImageUrl = item.data['image\_url'] ?? '';
* showDialog(
* context: context,
* builder: (context) {
* return AlertDialog(
* title: *const* Text('Chỉnh sửa sản phẩm'),
* content: SingleChildScrollView(
* child: Column(
* mainAxisSize: MainAxisSize.min,
* children: [
* TextField(
* controller: nameController,
* decoration: *const* InputDecoration(labelText: 'Tên sản phẩm'),
* ),
* TextField(
* controller: typeController,
* decoration: *const* InputDecoration(labelText: 'Loại sản phẩm'),
* ),
* TextField(
* controller: priceController,
* decoration: *const* InputDecoration(labelText: 'Giá sản phẩm'),
* keyboardType: TextInputType.number,
* ),
* *const* SizedBox(height: 10),
* \_image == null
* ? (existingImageUrl != ''
* ? Image.network(existingImageUrl, height: 100)
* : Text('Chưa có hình ảnh'))
* : Image.file(\_image!, height: 100),
* ElevatedButton(
* onPressed: \_pickImage,
* child: Text('Thay đổi hình ảnh'),
* ),
* ],
* ),
* ),
* actions: [
* ElevatedButton(
* onPressed: () async {
* String? imageUrl = existingImageUrl;
* if (\_image != null) {
* imageUrl = await \_uploadImage(\_image!);
* }
* DatabaseReference ref = FirebaseDatabase.instance.ref("products/${item.key}");
* await ref.update({
* 'name': nameController.text,
* 'type': typeController.text,
* 'price': int.parse(priceController.text),
* 'image\_url': imageUrl,
* });
* Navigator.of(context).pop();
* setState(() {
* \_image = null;
* });
* },
* child: *const* Text('Lưu'),
* ),
* ElevatedButton(
* onPressed: () {
* Navigator.of(context).pop();
* setState(() {
* \_image = null;
* });
* },
* child: *const* Text('Hủy'),
* ),
* ],
* );
* },
* );
* }
* }
* class Item {
* String key;
* Map<*dynamic*, *dynamic*> data;
* Item({*required* this.key, *required* this.data});
* }
* class ProductDetailPage extends StatelessWidget {
* *final* Item product;
* ProductDetailPage({*required* this.product});
* *@*override
* Widget build(BuildContext context) {
* var data = product.data;
* return Scaffold(
* appBar: AppBar(
* title: Text(data['name']),
* ),
* body: Padding(
* padding: *const* EdgeInsets.all(8.0),
* child: Column(
* children: [
* data['image\_url'] != null && data['image\_url'] != ''
* ? Image.network(data['image\_url'], height: 200)
* : Placeholder(fallbackHeight: 200),
* SizedBox(height: 10),
* Text(
* data['name'],
* style: TextStyle(fontSize: 24),
* ),
* Text('Loại: ${data['type']}'),
* Text('Giá: ${data['price']}'),
* ],
* ),
* ),
* );
* }
* }