



android

# ANDROID DEVELOPMENT

Banyapon Poolsawas



# Kotlin

# 5



Start from Scratch  
Web Application and Real-Time Database

**Banyapon Poolsawas**









Head of Interactive Design and Game Development,  
College of Creative Design and Entertainment Technology

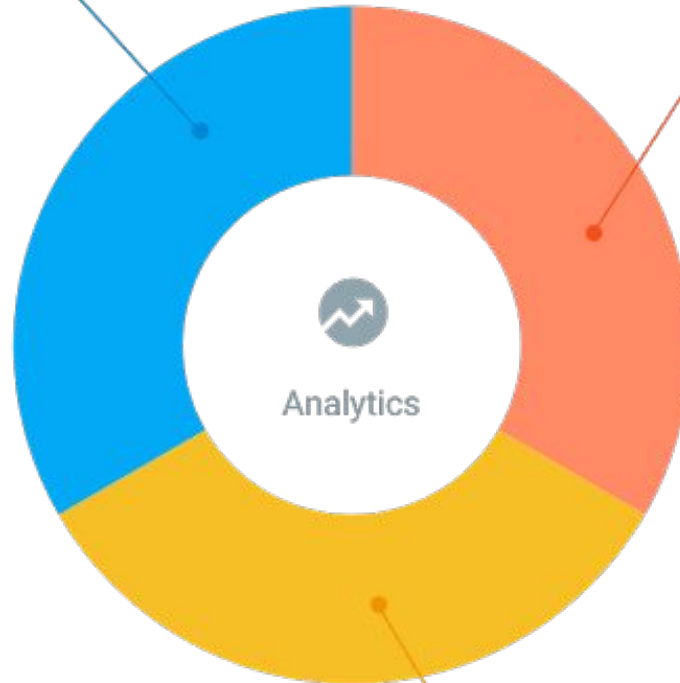
<https://www.facebook.com/ANTDPU/>

Environment






# ทำไมต้อง “FireBase” ?

## DEVELOP

-  Realtime Database
-  Authentication
-  Cloud Messaging
-  Storage
-  Hosting
-  Remote Config
-  Test Lab
-  Crash Reporting



## GROW

-  Notifications
-  App Indexing
-  Dynamic Links
-  Invites
-  AdWords

## EARN



# การสร้างเว็บไซต์ด้วย Firebase

ขั้นตอนที่ 1: เข้าไปที่เว็บไซต์ <https://firebase.google.com>

 **Firebase** Products Use Cases Pricing Docs Support

 Search

[GO TO CONSOLE](#)



Firebase helps you build better  
mobile apps and grow your  
business.

[GET STARTED](#)



[WATCH THE VIDEO](#)

## ขั้นตอนที่ 2:

เข้าระบบด้วย Google Account  
แล้วไปที่ Console หน้าจอจัดการ  
Projects ของเรา (มุมขวามือ)



## ขั้นตอนที่ 3:

สร้าง Project ใหม่ให้เรียบร้อย ตั้งชื่อให้เข้าใจ  
ง่าย แล้วคลิกเพื่อเริ่มใช้งาน

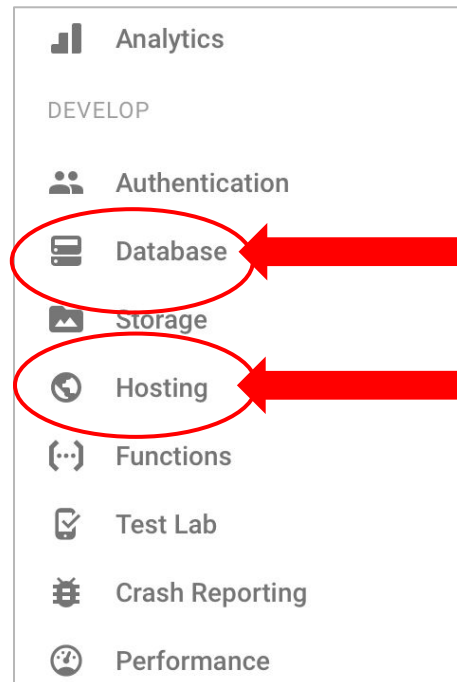
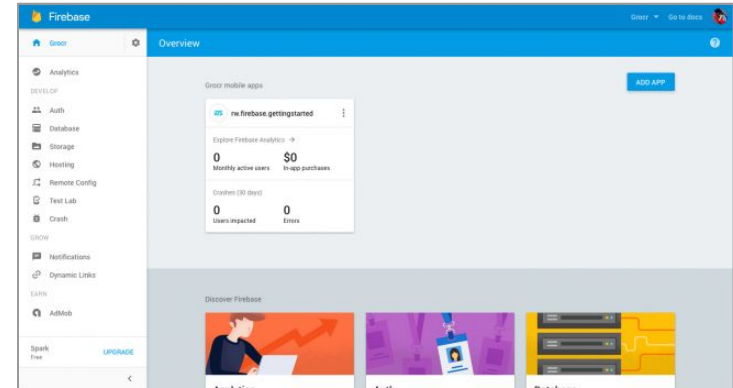
### Projects span apps across platforms.

A project is a container for your apps across platforms: Android, iOS and web. It supports sharing features such as Database, User Management and Remote Config across your apps.




## ขั้นตอนที่ 4:

พิจารณาเมนูของเครื่องมือที่  
FireBase ทุกตัวที่เมนู  
การใช้งานด้านข้าง



## ขั้นตอนที่ 5:

การทำงานของ Firebase จะอ้างอิง NodeJS ดังนั้นเราต้องมีการติดตั้ง NodeJS ก่อนให้ไปที่เว็บไซต์ <http://nodejs.org> ทำการดาวน์โหลด และติดตั้งให้เรียบร้อย



The screenshot shows the Node.js website homepage. At the top is the Node.js logo and a navigation bar with links: HOME, ABOUT, DOWNLOADS, DOCS, FOUNDATION, GET INVOLVED, SECURITY, and NEWS. Below the navigation bar is a large banner for "Node.js Interactive" with a cityscape background. The banner includes the text "Join us for Node.js Interactive happening in Vancouver, Canada October 4 - 6, 2017" and "NORTH AMERICA Vancouver, Canada October 4-6, 2017". Below the banner, there is a paragraph describing Node.js: "Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world." Below this paragraph, there is a section titled "Download for macOS (x64)" with two buttons: "v6.11.0 LTS Recommended For Most Users" and "v8.1.3 Current Latest Features". At the bottom, there are links for "Other Downloads", "Changelog", and "API Docs" for both versions, and a link to "Or have a look at the LTS schedule."

node

HOME | ABOUT | DOWNLOADS | DOCS | FOUNDATION | GET INVOLVED | SECURITY | NEWS

node JS Interactive  
NORTH AMERICA  
Vancouver, Canada  
October 4-6, 2017

Join us for Node.js Interactive  
happening in Vancouver, Canada  
**October 4 - 6, 2017**

Node.js® is a JavaScript runtime built on **Chrome's V8 JavaScript engine**. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, **npm**, is the largest ecosystem of open source libraries in the world.

Download for macOS (x64)

**v6.11.0 LTS**  
Recommended For Most Users

**v8.1.3 Current**  
Latest Features


Other Downloads | Changelog | API Docs   Other Downloads | Changelog | API Docs

Or have a look at the **LTS schedule**.

## ขั้นตอนที่ 6:

ในบทเรียนนี้จะเป็นการสร้าง Landing Page ที่มีการให้ข้อมูล และเก็บข้อมูลได้จากการกรอกฟอร์ม HTML เข้าฐานข้อมูลโดย Hosting จะใช้บริการของ Firebase และ Database ก็จะใช้บริการของ Firebase เช่นกัน

เริ่มต้นที่การติดตั้ง Hosting ก่อน เลือก Hosting คลิกที่ GET STARTED



### Hosting

Deploy web and mobile web apps in seconds using a secure global content delivery network

[Learn more](#) **GET STARTED**

## ขั้นตอนที่ 7:

เปิด Terminal หรือ Command Prompt ขึ้นมา ทำการเรียกคำสั่ง ดังนี้

```
$ npm install -g firebase-tools
```

เพื่อเป็นการติดตั้งเครื่องมือของ firebase สำหรับสร้าง Environment จาก NodeJS



## หน้าจอการติดตั้ง Firebase Tool ผ่าน Nodejs

```
www — -bash — 80x24
Last login: Thu Jun 29 20:23:10 on ttys000
[192:~ daydev$ npm
Usage: npm <command>

where <command> is one of:
[192:~ daydev$ ls
Desktop      Downloads    Movies       Pictures
Documents    Library      Music        Public
[192:~ daydev$ cd Desktop
[192:Desktop daydev$ mkdir www
[192:Desktop daydev$ cd www
[192:www daydev$ npm install -g firebase-tools
npm WARN deprecated node-uuid@1.4.8: Use uuid module instead
/usr/local/bin/firebase -> /usr/local/lib/node_modules/firebase-tools/bin/firebase
> grpc@1.3.8 install /usr/local/lib/node_modules/firebase-tools/node_modules/grpc
> node-pre-gyp install --fallback-to-build --library=static_library
[grpc] Success: "/usr/local/lib/node_modules/firebase-tools/node_modules/grpc/src/node/extension_binary/grpc_node.node" is installed via remote
```

ในตัวอย่างข้างต้น มีการสร้างโฟลเดอร์ www ไว้ที่ Desktop ของ OSX แล้วจึงเริ่มรันคำสั่ง install firebase-tool ผ่าน NodeJS

## ขั้นตอนที่ 8:

ทำการ Login เข้า Firebase ผ่าน firebase-tool ใน command line ด้วยคำสั่ง

```
$ firebase login
```

```
eyes@0.1.8
pkginfo@0.3.1
stack-trace@0.0.10
Sign in to Google:
[192:www daydev$ firebase login
? Allow Firebase to collect anonymous CLI usage information? (Y/n)
```

ระบบจะถามว่า เราจะอนุญาตให้ Firebase จัดการข้อมูลผ่าน Command Line ใหม่ให้ตอบ “Y” เราก็ให้สิทธิการจัดการผ่าน Email ของเราเลย




### Choose an account

to continue to **Firebase CLI**




Banyapon Poolsawas  
banyapon.poo@dpu.ac.th







# หน้าจอการให้สิทธิ Firebase ผ่าน Command Line ให้ทำการ Allow ยอมรับให้เรียบร้อย



## Hi Banyapon

 banyapon.poo@dpu.ac.th

**Firebase CLI** wants to

-  View and manage your data across Google Cloud Platform services 
-  View your Cloud Platform projects 
-  View and administer all your Firebase data and settings 

**Allow Firebase CLI to do this?**

By clicking Allow, you allow this app to use your information in accordance to their terms of service and privacy policies. You can remove this or any other app connected to your account in [My Account](#)

CANCEL ALLOW

Woohoo!

## Firebase CLI Login Successful

You are logged in to the Firebase Command-Line interface. You can immediately close this window and continue using the CLI.

เราสามารถรันคำสั่ง Firebase ผ่าน command line ได้สมบูรณ์แล้ว

```
Visit this URL on any device to log in:
https://accounts.google.com/o/oauth2/auth?client_id=563584335869-fgrhmd47bnekj5i8b5pr03ho849e6.apps.googleusercontent.com&scope=email%20openid%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloudplatformprojects.readonly%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Ffirebase%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform&response_type=code&state=807616673&redirect_uri=http%3A%2F%2Flocalhost%3A9005

Waiting for authentication...

✓ Success! Logged in as banyapon.poo@dpu.ac.th
192:www daydev$
```

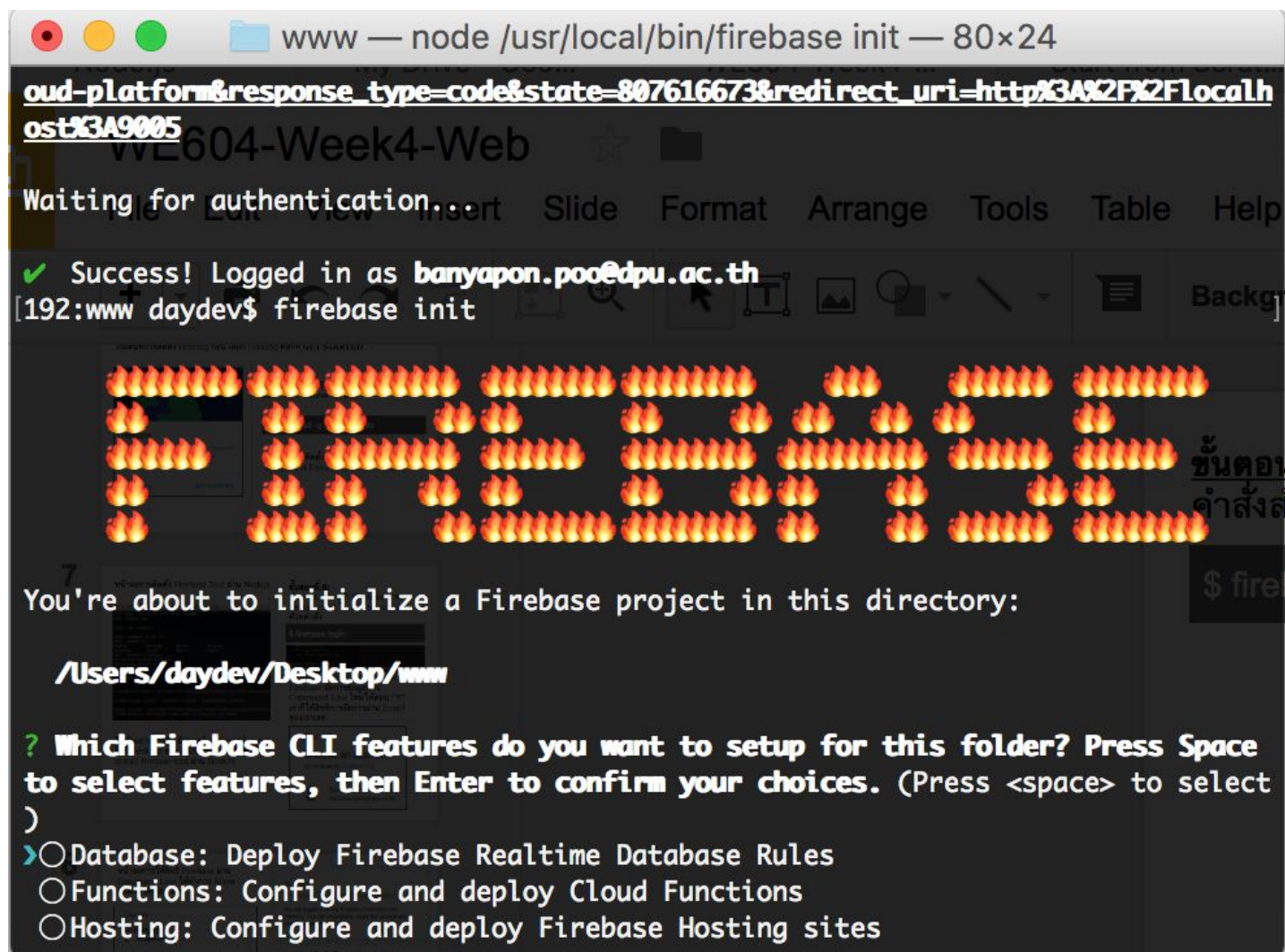
ขั้นตอนต่อไปคือ การ initial ตัว Project ให้อยู่ใน Environment ของ Firebase ผ่าน Command Line โดยจะมีการสร้าง Virtual Directory สำหรับทำการ Deploy ซึ่ง Concept จะคล้ายการทำ Git

## ขั้นตอนที่ 9:

คำสั่งสำหรับ initial project คือ:

```
$ firebase init
```

ตัวเลือกที่เราต้องการทำงานนั้นจะปรากฏ 3 ตัว  
เลือก คือ Database, Functions และ Hosting  
ให้ เลือกโดยการกด Space Bar ซึ่งสามารถเลือก  
ได้ทั้ง 3 รายการ  
ในขั้นตอนนี้ ให้เลือก Database และ Hosting



```
www — node /usr/local/bin/firebase init — 80x24
oud-platform&response_type=code&state=807616673&redirect_uri=http%3A%2F%2Flocalh
ost%3A9005
WVL604-Week4-Web
Waiting for authentication.
✓ Success! Logged in as banyapon.poo@dpu.ac.th
[192:www daydev$ firebase init

  ____  _____  _____  _____  _____  _____  _____  _____
 / ___/  / ___/  / ___/  / ___/  / ___/  / ___/  / ___/  / ___/  / ___/  / ___/
/_/    /_/    /_/    /_/    /_/    /_/    /_/    /_/    /_/    /_/    /_/

You're about to initialize a Firebase project in this directory:

/Users/daydev/Desktop/www

? Which Firebase CLI features do you want to setup for this folder? Press Space
to select features, then Enter to confirm your choices. (Press <space> to select
)
> ☐ Database: Deploy Firebase Realtime Database Rules
  ☐ Functions: Configure and deploy Cloud Functions
  ☐ Hosting: Configure and deploy Firebase Hosting sites
```



## ตัวอย่างการเลือก Database และ Hosting ใน Command Line ของ Firebase

```
? Which Firebase CLI features do you want to setup for t
to select features, then Enter to confirm your choices.
  ● Database: Deploy Firebase Realtime Database Rules
  ○ Functions: Configure and deploy Cloud Functions
  > ● Hosting: Configure and deploy Firebase Hosting sites
```

กด Enter แล้วทำการเลือกชื่อ Project ที่เราสร้างไว้บนเว็บไซต์ Firebase

```
== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Select a default Firebase project for this directory:
  [don't setup a default project]
  > WE604Webs (we604webs)
    AutoStreetView (autostreetview)
    RestMeIn (restmein)
    DaydevProject (daydevproject)
    [create a new project]
```

กด Space Bar ที่ชื่อ Project ที่เราต้องการ Setup ระบบจะทำการถามว่าเราต้องการ Deploy ผ่าน folder ชื่ออะไร

ส่วนของ Database เป็นการเก็บค่า Rule และ Config Json ไว้ใน Directory ที่เราต้องการเก็บ ใช้ชื่อ Default ได้

```
? Select a default Firebase project for this directory: WE604Webs (we604webs)

== Database Setup

Firebase Realtime Database Rules allow you to define how your data should be
structured and when your data can be read from and written to.

? What file should be used for Database Rules? (database.rules.json)
```

ส่วนของ Hosting ให้เรากำหนด Folder ที่จะเป็น Virtual Directory ในตัวอย่างกำหนดชื่อว่า “public”

```
== Hosting Setup

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? (public) public
```

ให้สร้าง single file ตัว index.html ไปเลย

```
8
i Writing configuration info to firebase.json...
i Writing project information to .firebase...

✓ Firebase initialization complete!
192:www daydev$
```

## ขั้นตอนที่ 10:

สำหรับการ Deploy ตัวเว็บไซต์ขึ้นไป  
Live ให้เป็นเว็บไซต์จริงๆ ใช้คำสั่ง

```
$ firebase deploy
```

การทดสอบให้ดู Log ในระบบบนหน้า  
Console ของเว็บไซต์ Firebase  
ตัวอย่าง:

<https://we604webs.firebaseio.com>

### Domain

CONNECT DOMAIN

Domain

Type

Status

we604webs.firebaseio.com

Default

### Deployment history

Status

Time

Deploy

Files



Deploying

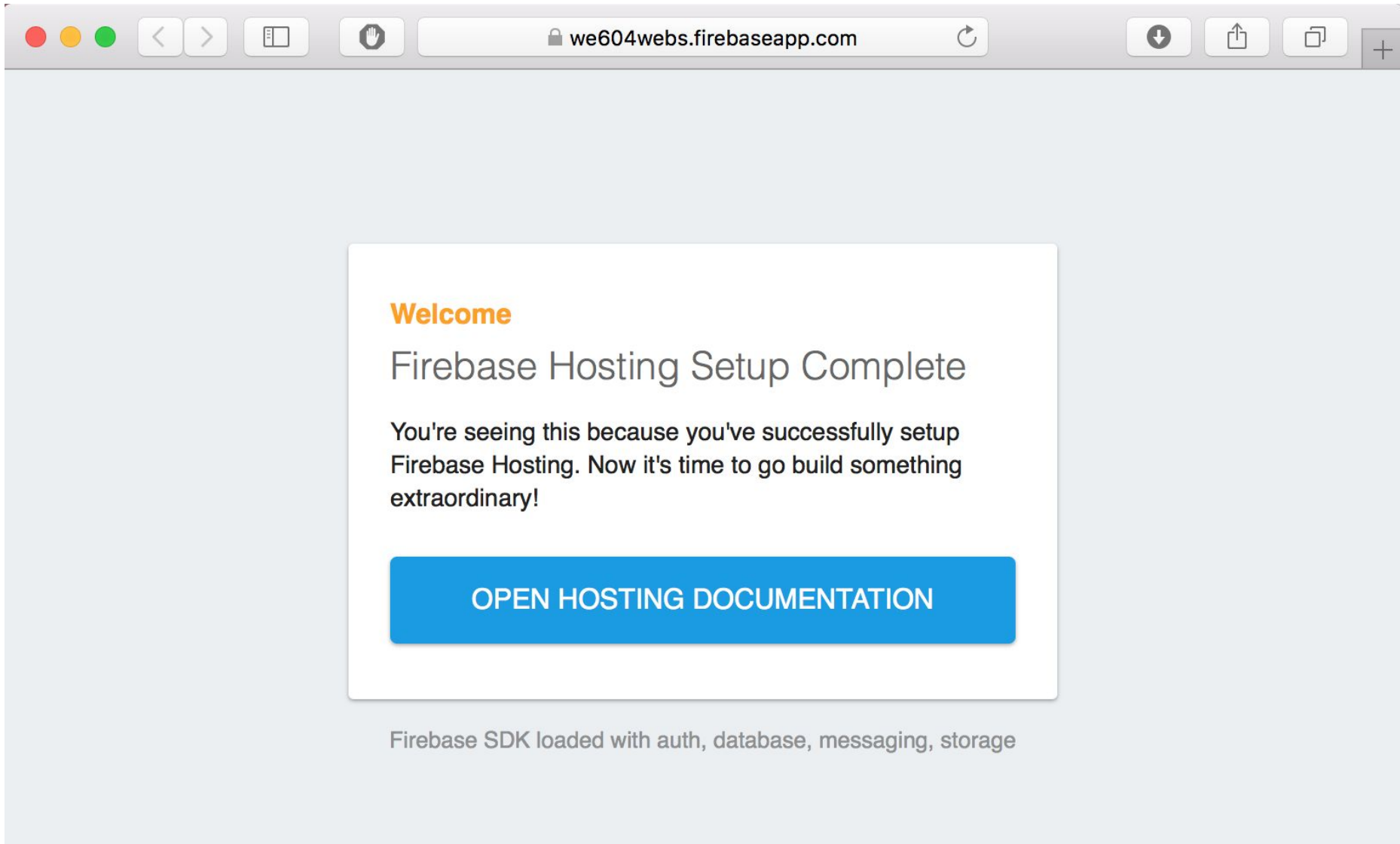
Rows per page: 10



1-1 of 1



เว็บไซต์แรกของคุณ:

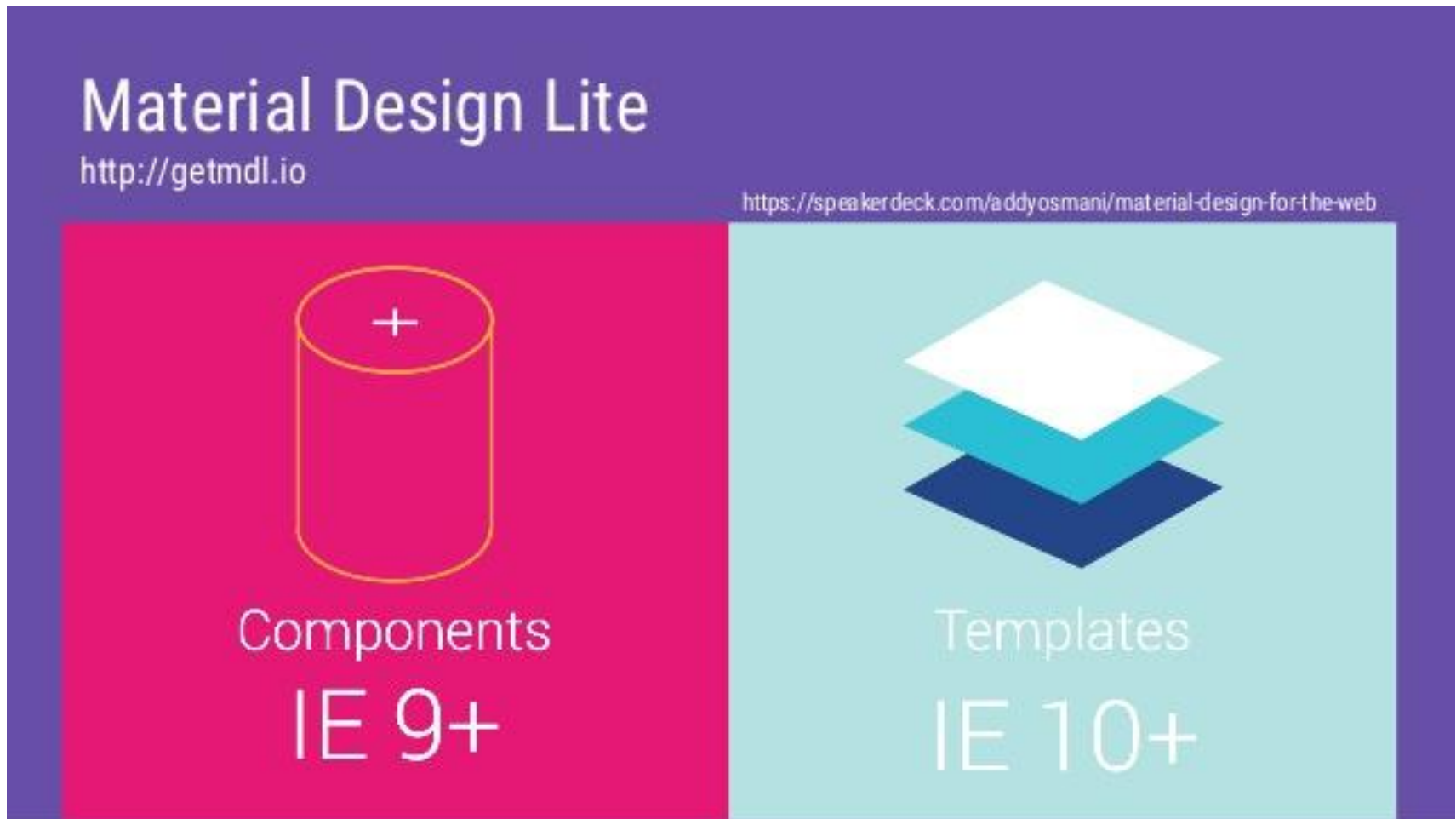


หมายเหตุ: เราสามารถจำลอง Localhost ได้ด้วยคำสั่ง

\$ firebase serve

Design

สำหรับการออกแบบ UI สามารถใช้ CSS  
หรือ Framework พวก Bootstrap สำหรับออกแบบได้  
แต่ในตัวอย่างนี้จะใช้ UI Framework ที่เรียกว่า **Material Design Lite (MDL)**



เว็บไซต์: <http://getmdl.io>

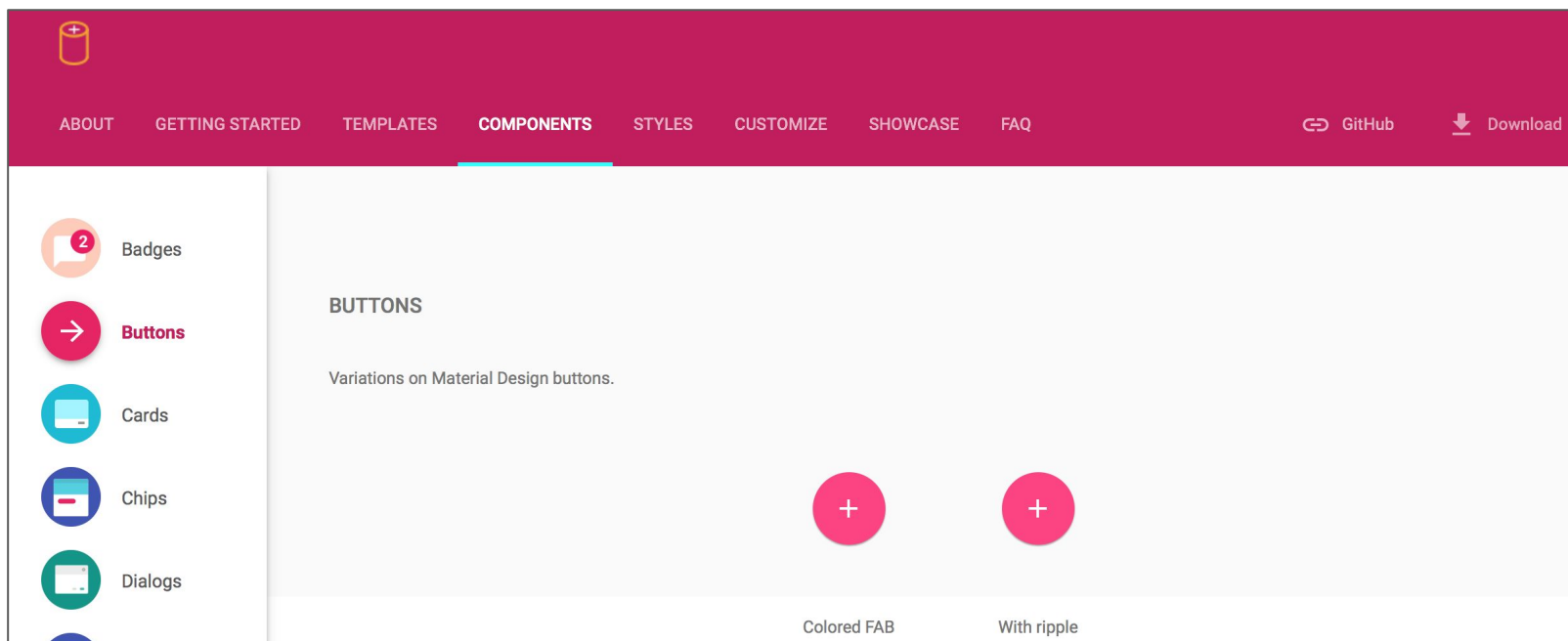


## ขั้นตอนที่ 11:

เปิดไฟล์ index.html จาก firebase ที่เราสร้างขึ้นแล้วทำการ setup Host ของ CSS ตัว MDL ดังนี้: <https://getmdl.io/started/index.html> ใน Tag ส่วน <HEAD>

```
<link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">  
<link rel="stylesheet" href="https://code.getmdl.io/1.3.0/material.indigo-pink.min.css">  
<script defer src="https://code.getmdl.io/1.3.0/material.min.js"></script>
```

ส่วนของการออกแบบหน้าเว็บไซต์ด้วย MDL นั้นให้ไปที่ URL: <https://getmdl.io/components/index.html> เพื่อศึกษาการทำงานของ MDL



ตัวอย่างสำหรับหน้า Landing Page ที่เหมาะสม  
วัตถุประสงค์ จะทำหน้าสมัครสมาชิกที่เก็บข้อมูล ชื่อ,นามสกุล,อีเมล,ที่อยู่ ผ่าน Firebase  
แทรก Code ชุดนี้ใน <body>

```
<div class="mdl-layout mdl-js-layout mdl-layout--fixed-header">
  <header class="mdl-layout__header">
    <div class="mdl-layout__header-row">
      <!-- Title -->
      <span class="mdl-layout-title">ANT DPU</span>
      <div class="mdl-layout-spacer"></div>
      <nav class="mdl-navigation mdl-layout--large-screen-only">
        <a class="mdl-navigation__link" href="http://www.facebook.com/ANTDPU">
          ANT DPU on Facebook
        </a>
      </nav>
    </div>
  </header>
  <div class="mdl-layout__drawer">
    <span class="mdl-layout-title">Title</span>
    <nav class="mdl-navigation">
      <a class="mdl-navigation__link" href="http://www.facebook.com/ANTDPU">
        ANT DPU on Facebook
      </a>
    </nav>
  </div>
  <main class="mdl-layout__content">
    <div class="page-content"><!-- Your content goes here -->
      <p id="contact-success" class="text-success lead"></p>
      <p id="contact-error" class="text-danger lead"></p>
    </div>
  </main>
```

แทรก Code ชุดนี้ในตำแหน่งต่อจาก <!-- Your content hoes here -->

```
<!-- Your content goes here →
<p id="contact-success" class="text-success lead"></p>
<p id="contact-error" class="text-danger lead"></p>
<div align="center" style="padding-top: 20px;">
<form id="contact-form" role="form" onsubmit="return sendData(this)">
  <table class="mdl-data-table mdl-js-data-table">
    <thead>
      <tr><td>NAME:</td>
        <td><div class="mdl-textfield mdl-js-textfield">
          <input class="mdl-textfield__input" type="text" id="name" name="name">
          <label class="mdl-textfield__label" for="name">Your Full Name...</label>
        </div></td></tr>
      <tr><td>Email:</td>
        <td><div class="mdl-textfield mdl-js-textfield">
          <input class="mdl-textfield__input" type="email" id="email" name="email">
          <label class="mdl-textfield__label" for="email">Your Email Address...</label>
        </div></td></tr>
      <tr><td>Message:</td>
        <td><div class="mdl-textfield mdl-js-textfield">
          <textarea class="mdl-textfield__input" type="text" rows="4" id="message" name="message"></textarea>
          <label class="mdl-textfield__label" for="address">Your Message...</label>
        </div>
        <br/>
        <button class="mdl-button mdl-js-button mdl-button--raised mdl-js-ripple-effect" id="send-data" type="submit"
>Submit Data</button>
      </td></tr>
    </thead>
    <tbody>
    </tbody>
  </table>
</form>
</div>
```

หน้าจอบริษัทผลลัพธ์จะเป็นดังนี้:

The image shows a web browser window with the address bar set to 'localhost'. The page has a blue header with a hamburger menu icon and the text 'ANT DPU'. Below the header is a form with three input fields and a submit button.

**NAME:**

**Email:**

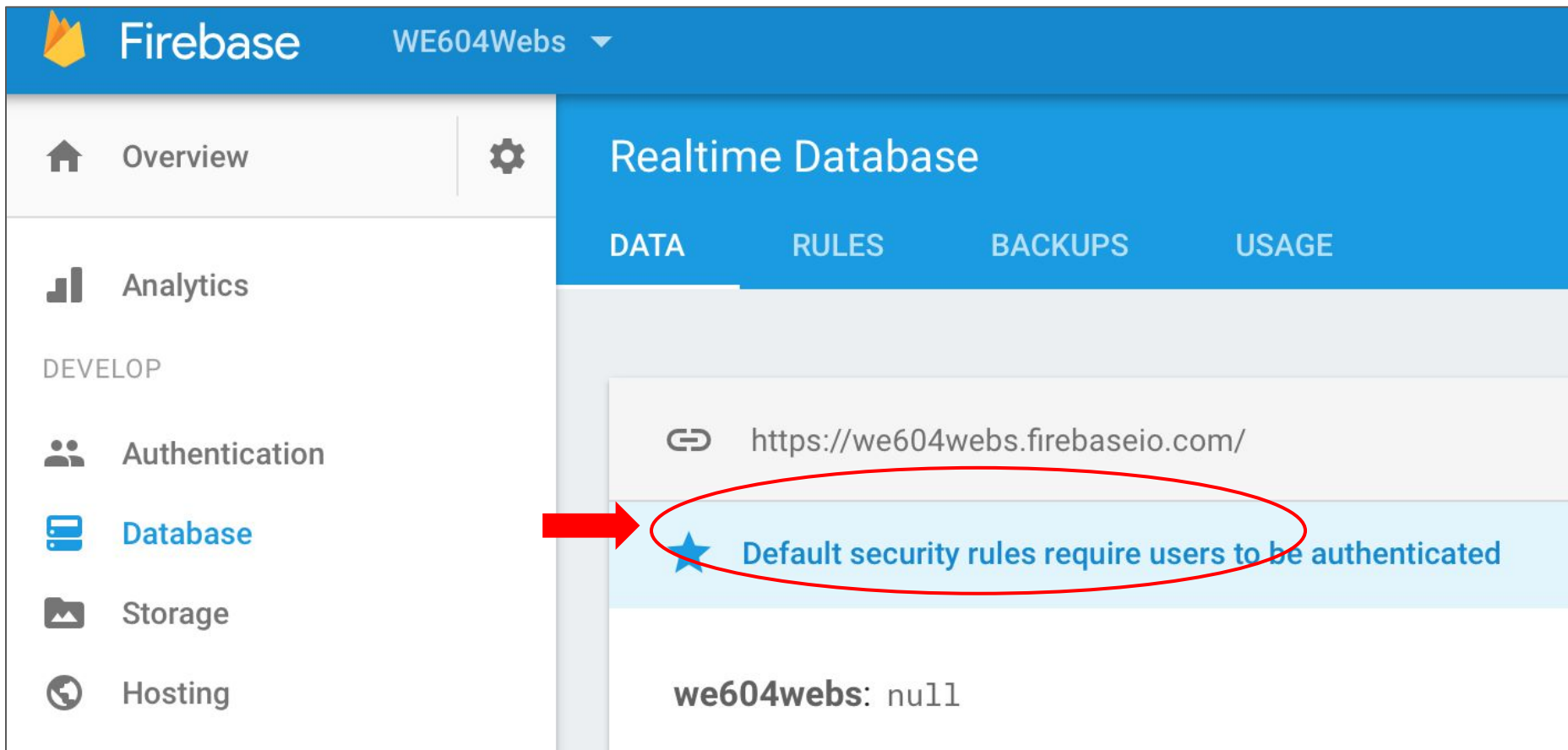
**Address:**

**SUBMIT DATA**

Database

## ขั้นตอนที่ 12:

ไปที่เว็บไซต์ firebase เลือกเมนู Database ทำการคัดลอก URL ของ Database เราไว้ในตัวอย่างคือ: <https://we604webs.firebaseio.com/>



## ขั้นตอนที่ 13:

ดาวน์โหลดไฟล์ firebase.js ไปวางไว้ที่ public/js/ ที่เราสร้างไว้ ดาวน์โหลดที่:

<https://drive.google.com/file/d/OB08PZSOd4UmOLVQ5NXdaX1FudGc/view?usp=sharing>

### ขั้นตอนที่ 13:

เปิด Firebase Database ของเราโดยการกำหนด Rule ให้เป็น public แก้ไขดังนี้:



The screenshot shows the Firebase Rules editor interface. At the top, there is a blue header bar with the text "Unpublished changes" on the left, a "PUBLISH" button in the center, and a "DISCARD" button on the right. Below the header, there is a text area containing the following JSON rule configuration:

```
1 {  
2   "rules": {  
3     ".read": true,  
4     ".write": true  
5   }  
6 }  
7
```

### ขั้นตอนที่ 14:

แทรก JS Script คำสั่งต่อไปนี้บน <head>:

```
<script defer src="js/firebase.js"></script>
```

## ขั้นตอนที่ 15:

แทรก JS Script สำหรับฟังก์ชัน SendData() ไว้ก่อนปิด </body> เป็น JS สำหรับทำงานร่วมกับ Firebase Database แทรกแทนค่าชื่อ database ลงไป ดังนี้:

```
<script>
var contactFrom = document.getElementById('contact-form');
var contactSuccess = document.getElementById('contact-success');
var contactError = document.getElementById('contact-error');
var sendBtn = document.getElementById('send-data');
var onMessageComplete = function(error) {
    sendBtn.disabled = false;
    if (error) {
        contactError.innerHTML = 'Sorry. Could not send Data.';
    } else {
        contactSuccess.innerHTML = "Success !";
        // hide the form
        contactFrom.style.display = 'none';
    }
};
function sendData(formObj) {
    var myFirebaseRef = new Firebase("https://yourappname.firebaseio.com/data");
    myFirebaseRef.push({
        name: formObj.name.value,
        email: formObj.email.value,
        message: formObj.message.value
    }, onMessageComplete);
    sendBtn.disabled = true;
    return false;
}
</script>
```



ทดสอบโดยการ Deploy ขึ้นเว็บจริง (หรือทดสอบก็ได้) กรอก ข้อมูลหน้าเว็บไซต์ ให้ครบดังนี้:

## ข้อมูลถูกเก็บแบบ Real-Time Database แล้ว

The screenshot shows the Realtime Database interface. At the top, there are tabs for DATA, RULES, BACKUPS, and USAGE. The URL bar shows the database path: `https://we604webs.firebaseio.com/`. Under the `we604webs` root, there is a `data` node. Inside `data`, there is a node `-KnoloKNRfUWwkt1V61u`. This node contains a JSON object with the following fields:

- `email`: `"banyapon@daydev.com"`
- `message`: `"ทดสอบ โอเคนะ"`
- `name`: `"Banyapon Poolasawas"`



android

THANK YOU  
FOR YOUR  
ATTENTIONS