

**كلية  
الحاسبات والمعلومات**



**Faculty of computer and Informatics**

**Menofia University**

**2017/2018**

**IFood**

**IOS Platform)(**

**Sponsored**

**BY**

**DR. Amira Ibrahim**

---

*About book*

---

**A Graduation Project Report Submitted to  
The Faculty of computer and Informatics Menofia  
University  
In Partial Fulfillment of the Requirements for the  
Degree of  
diploma of computer science  
In  
Faculty of computer and Informatics, Menofia  
University  
Menofia, Egypt  
July 2018**

---

## *Teamwork*

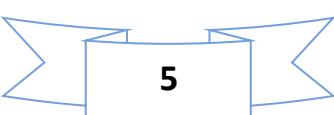
---

- **Mohammed Arafa Ebeid Aly.**
- **Aly fawzy Mohammed Ahmed.**
- **Aya Adel Elnagar**

## Table of Contents

Contents	Page
<b>Chapter one "Introduction"</b> .....	6
1.0 About Dr Amira Ibrahim .....	7
1.1 IFood .....	8
1.2 Aim of the work .....	8
1.3 IFood Methodology .....	9
<b>Chapter two "Technologies"</b> .....	10
2.1 IOS Platform .....	11
2.2 Json .....	13
2.3 GitHub .....	14
2.4 firebase .....	19
2.5 API .....	24
<b>Chapter Three “Application Design”</b> .....	26
3.1 Design Interface .....	27
3.1.1 buttons .....	28
3.1.2 labels .....	29
3.1.3 Text Field .....	29
3.1.4 Image View .....	29
3.1.5 Search Bar .....	30
3.1.6 Table View .....	30
3.1.7 Stare Cosmos .....	31
3.1.8 Collection View .....	31
3.2 IFood App Design user login .....	33
3.2.1 Skip button .....	34
3.2.2 IFood image view .....	34
3.2.3 username and password text field .....	34
3.2.4 Login button .....	35
3.2.5 Reset password label with tap gesture action .....	35
3.3 Reset Password .....	36

3.3.1 Skip button.....	37
3.3.2 IFood image view.....	37
3.3.3 Email Text field .....	37
3.3.4 Reset pass Button.....	38
3.3.5 Sign in label with tap gesture action .....	38
3.4 sign up .....	39
3.4.1 Skip button.....	40
3.4.2 User Name, Email, Password and Confirm Password Text fields.....	40
3.4.3 Sign Up Button.....	41
3.4 Home Page .....	42
3.4.1 Search Bar .....	43
3.4.2 Table View.....	43
3.5 Restaurant Page .....	47
3.5.1 Skip button.....	48
3.5.2 Restaurant Image View, Restaurant name Label, Restaurant Location Label, Restaurant phone Label, Restaurant Rate Cosmos Stare.....	48
3.5.3 Restaurant Food button. ....	49
3.6 Food page .....	50
3.6.1 Skip button.....	51
3.6.2 Collection View .....	51
<b>Chapter Four "Sign In Validation Using FireBase".....</b>	<b>55</b>
4.1 What is cocoaPods .....	56
4.1.1 install cocoaPods.....	56
4.1.2 connect App to Firebase.....	57
<b>Chapter five “connect Application to server”.....</b>	<b>62</b>
5.1 GitHub .....	63
5.2 Json Server .....	71
5.3 Finally Models and Extensions .....	73
5.3.1 Models.....	73
5.3.2 Extensions.....	74
<b>References .....</b>	<b>78</b>



# **Chapter one**

# **"Introduction"**

## **1.0 About Dr Amira Ibrahim.**

**Name:** Amira Ibrahim.

**College:** faculty of computer and informatics.

**University:** Menofia University.

**The field of excellence:** Lecturer at Faculty of Computer & Information - Menoufia University and Assistant Lecturer at Faculty of Computers and Information at menofia university.

## **1.1 IFood**

This is an IOS Application used to display all restaurants in Zagazig surrounding the user

The Application has the potential to contain an infinite number of restaurants.

These restaurants are arranged in the application with the user feedback rating.

Each restaurant has its own rating, detailed information and a special menu that is characterized by its characteristic

Each user will choose the restaurant they like, the food they want, and contact the restaurant through restaurant information.

## **1.2 Aim of the work**

In this study, we aim to develop an application toolbox to facilitate the aim of the application is to make it easier for the user to know the best restaurants in the Zagazig area and the best food served by these restaurants. It also saves time, effort and money in order to reach these restaurants. The program also makes propaganda for these restaurants, thus supporting these restaurants to compete among themselves to reach the best evaluation through the quality of food and good dealing with customers.

## **1.3 IFood Methodology**

A way to develop the application from start to end by four steps:

### **1) Design:**

To design any IOS application we should use IOS platform

### **2) Create Backend:**

in this step backend created JavaScript Object Notation (Json)code.

### **3) Sign in validation:**

to manage any user sign in IFood application we need to database using firebase service.

### **4) Connect to server:**

First upload my json code into GitHub as a server

Second how to use the code from GitHub to my application by using (my-json-server).

# **Chapter two**

# **"Technologies"**

## 2.1 IOS Platform

iOS (formerly iPhone OS) is a mobile operating system created and developed by Apple Inc. exclusively for its hardware. It is the operating system that presently powers many of the company's mobile devices, including the iPhone, iPad, and iPod Touch.

You can test many of the Cordova features using the iOS simulator installed with the iOS SDK and XCode, but you need an actual device to fully test all of the app's device features before submitting to the App Store. The device must have at least iOS 8 installed, the minimum iOS version supported as of Cordova 4.0.0. Supported devices include iPhone 4S, iPhone 5, iPhone 5C, iPhone 5S, iPhone 6, iPhone 6 Plus, iPhone 6S, iPhone 6S Plus, iPhone SE, iPad 2, iPad 3, iPad 4, iPad Air, iPad Air 2, iPad Pro, iPad Mini, iPad Mini 2, iPad Mini 3, iPod Touch 5th gen and iPod Touch 6th gen or later.

### Project Configuration

Installing XCode will mostly set everything needed to get started with the native side of things. You should now be able to create and build a Cordova project. For more details on installing and using the CLI, refer to Create your first app guide.

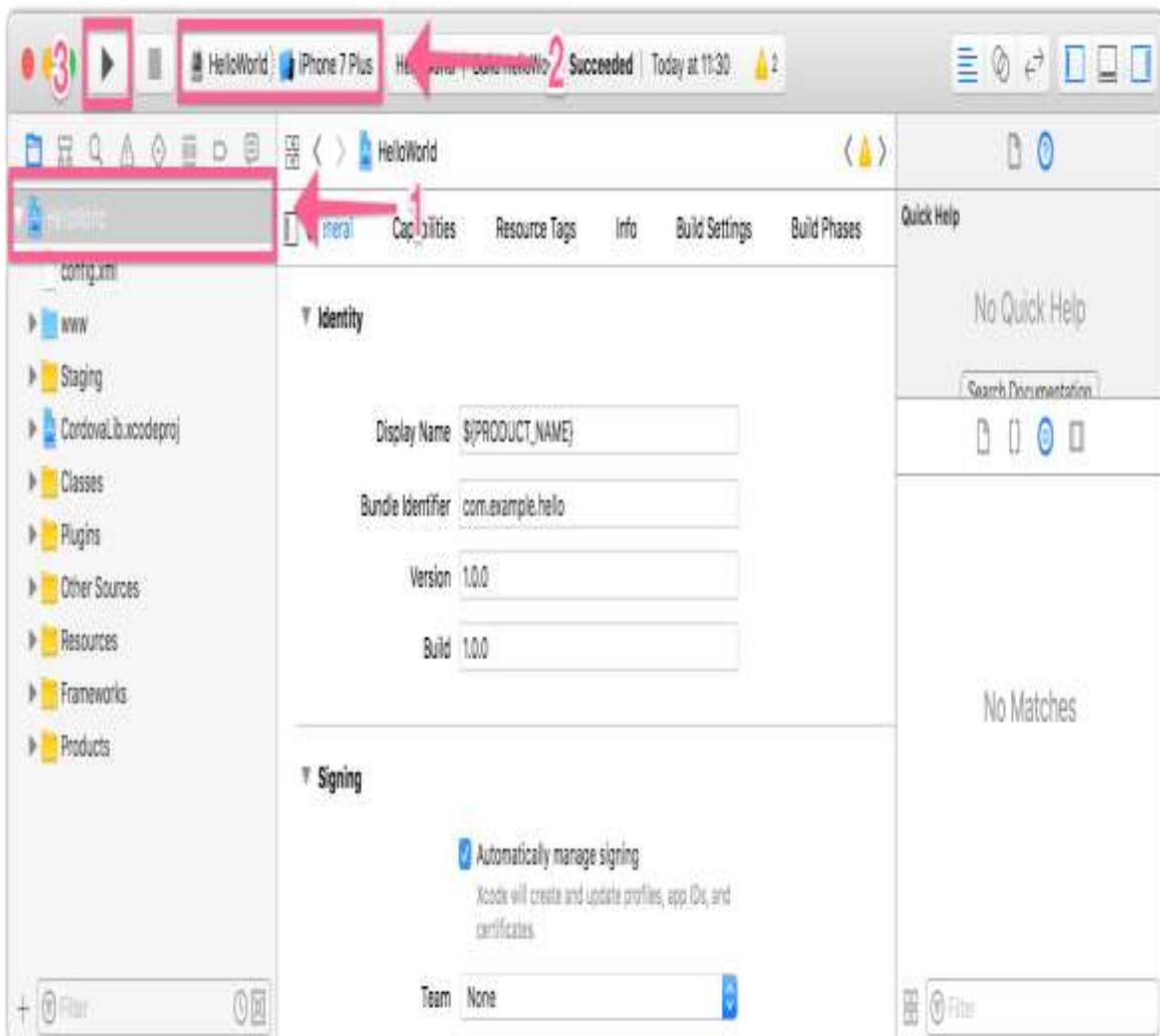
### Deploying to Simulator

To preview the app in the iOS simulator:

1. Open the workspace file (platforms/ios/HelloWorld.xcworkspace) from XCode, or from the command line:

2. \$ open ./platforms/ios/HelloWorld.xcworkspace/

3. Make sure the HelloWorld project is selected in the left panel (1).



1. Select the intended device from the toolbar's **Scheme** menu, such as the iPhone 7 Plus Simulator as highlighted in (2)
2. Press the **Run** button (3) in the same toolbar to the left of the **Scheme**. That builds, deploys, and runs the application in the simulator. A separate simulator application opens to display the app:

## 2.2 Json

The JSON format is often used for serializing and transmitting structured data over a network connection. It is used primarily to transmit data between a server and web application, serving as an alternative to XML. JSON is JavaScript Object Notation.

### Storing JSON Data

As a simple example, information about me might be written in JSON as follows:

```
var jason = {  
    "age" : "24",  
    "hometown" : "Missoula, MT",  
    "gender" : "male"  
};
```

This creates an object that we access using the variable `jason`. By enclosing the variable's value in curly braces, we're indicating that the value is an object. Inside the object, we can declare any number of properties using a "name": "value" pairing, separated by commas. To access the information stored in `jason`, we can simply refer to the name of the property we need.

With the rise of AJAX-powered sites, it's becoming more and more important for sites to be able to load data quickly and asynchronously, or in the background without delaying page rendering. Switching up the contents of a certain

element within our layouts without requiring a page refresh adds a "wow" factor to our applications, not to mention the added convenience for our users. Because of the popularity and ease of social media, many sites rely on the content provided by sites such as Twitter, Flickr, and others. These sites provide RSS feeds, which are easy to import and use on the server-side, but if we try to load them with AJAX, we run into a wall: we can only load an RSS feed if we're requesting it from the same domain it's hosted on. An attempt to load my Flickr account's RSS feed via jQuery's `$.ajax()` method results in the following JavaScript error:

One of the easiest ways to load JSON data into our web applications is to use the `$.ajax()` method available in the jQuery library. The ease of retrieving data will vary based on the site providing the data.

## 2.3 GitHub

It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project. GitHub offers plans for both private repositories and free accounts which are commonly used to host open-source software projects.

GitHub is a website and service that we hear geeks rave about all the time, yet a lot of people don't really understand what it does. Want to know what all the GitHub hubbub is about? Read on to find out.

## The “Git” in GitHub

### RELATED: Version Tracking with Subversion (SVN) For Beginners

To understand GitHub, you must first have an understanding of Git. Git is an open-source version control system that was started by Linus Torvalds—the same person who created Linux. Git is similar to other version control systems—Subversion, CVS, and Mercurial to name a few.

So, Git is a version control system, but what does that mean? When developers create something (an app, for example), they make constant changes to the code, releasing new versions up to and after the first official (non-beta) release.

Version control systems keep these revisions straight, storing the modifications in a central repository. This allows developers to easily collaborate, as they can download a new version of the software, make changes, and upload the newest revision. Every developer can see these new changes, download them, and contribute.

Similarly, people who have nothing to do with the development of a project can still download the files and use them. Most Linux users should be familiar with this process, as using Git, Subversion, or some other similar method is pretty common for downloading needed files—especially in preparation for compiling a program from source code (a rather common practice for Linux geeks).

Git is the preferred version control system of most developers since it has multiple advantages over the other systems available. It stores file changes more efficiently and ensures

file integrity better. If you're interested in knowing the details, the Git Basics page has a thorough explanation on how It works.

### The “Hub” in GitHub

We've established that Git is a version control system, similar but better than the many alternatives available. So, what makes GitHub so special? Git is a command-line tool, but the center around which all things involving Git revolve is the hub—GitHub.com—where developers store their projects and network with likeminded people.

Let's go over a few of the main reasons that geeks like to use GitHub and learn some terminology along the way.

### How to Create an Account on GitHub

GitHub is an online; browser based distributed version control system for software developers using the Git revision control system. The service provides free public repositories, issue tracking, graphs, code review, downloads, wikis, collaborator management, and more. There are also social networking elements of the service including feeds, the ability to follow other users, and network related graphs. This article will walk you through the process of signing up for GitHub.

## Steps

1

Go to the GitHub sign up page.

Username

This will be your username — you can enter your organization's username next.

Email Address

You will occasionally receive account related emails. We promise not to share your email with anyone.

Password

Use at least one lowercase letter, one numeral, and seven characters.

By clicking on "Create an account" below, you are agreeing to the [Terms of Service](#) and the [Privacy Policy](#).

 [Create an account](#)

2

Enter a username, valid email address, and password. Use at least one lowercase letter, one numeral, and seven characters.

3

Review carefully the GitHub Terms of Service and Privacy Policy before continuing. Upon clicking the “Create an account” button you will simultaneously be agreeing to these documents.

## Choose your personal plan

- Unlimited public repositories for free.
- Unlimited private repositories for \$7/month.

Don't worry, you can cancel or upgrade at any time.

### Help me set up an organization next

Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees.  
[Learn more about organizations.](#)

→ **Continue**

# 4

Choose a plan. You have two choices: Free and paid, the paid version has private repositories with \$7/month. You should try the free version then have the suitable choice.

 Completed  
Set up a personal account

 Step 2:  
Choose your plan

 Step 3:  
Tailor your experience

How would you describe your level of programming experience?

- Totally new to programming       Somewhat experienced       Very experienced

What do you plan to use GitHub for? (check all that apply)

- Design       Development       Project Management  
 School projects       Research       Other (please specify)

Which is closest to how you would describe yourself?

- I'm a student       I'm a hobbyist       I'm a professional  
 Other (please specify)

What are you interested in?

e.g. tutorials, android, ruby, web-development, machine-learning, open-source

[Sign in now to](#)

 **Submit**      skip this step 

**5**

Tailor experience. If you have time, fill in the survey or skip it.

**6**

You finished! Your GitHub account created!

## 2.4 firebase

Firebase is a mobile-backend-as-a-service that provides powerful features for building mobile apps. Firebase has three core services: a Realtime database, user authentication and hosting. With the Firebase iOS SDK, you can use these services to create apps without writing any server code. In this Firebase tutorial, you'll learn the fundamentals of Firebase by making a collaborative grocery list app called *Grocr*. When items get added to the list, they'll appear instantly on any user's devices, but you're not going to stop there. You'll tweak *Grocr* to work offline, so the list stays in sync even with a spotty grocery store data connection.

As you work, you'll learn about:

- Saving data to a Firebase database.
- Syncing data in real time.
- Authenticating users.
- Monitoring online users.
- Enabling offline support.

Get ready to real time all the things!

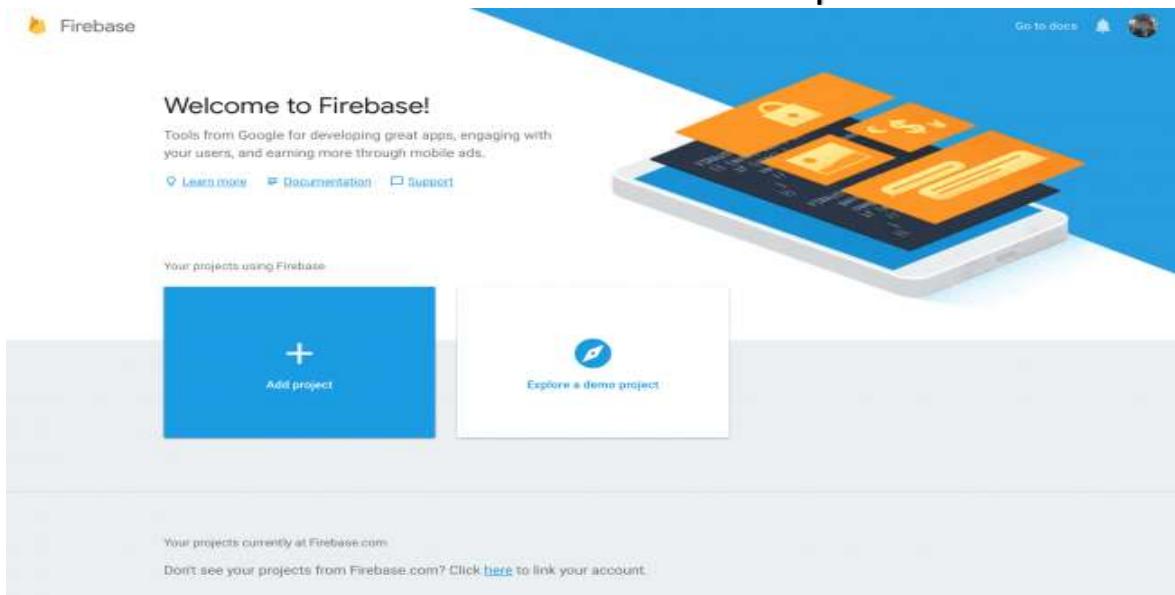
## Setting up a Firebase Account

There are three main steps to setting up Firebase in an iOS project:

1. Create a free Firebase account
2. Download and add *GoogleService-Info.plist* to your app
3. Tell Firebase to start when your app launches

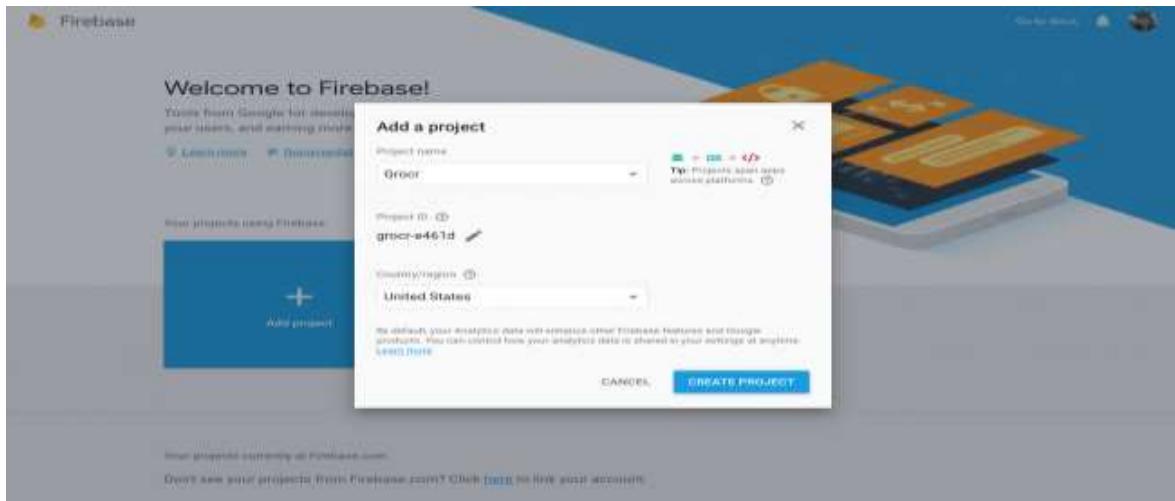
To create a Firebase account, visit the Firebase homepage. Press *GO TO CONSOLE* in the top-right corner, and enter the credentials for your Google account, if you're not signed in already. If you don't have a Google account, you'll need to create one first, which you can do here.

You'll then have a clean Firebase console created for you. Don't worry about forking over any money; everything in this Firebase tutorial can be done with the free plan.

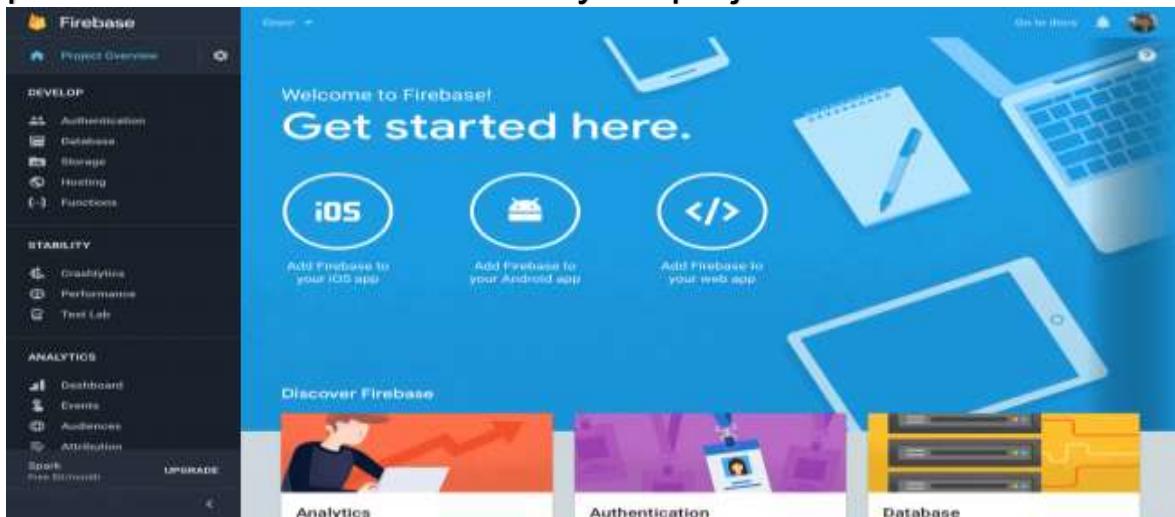


It's time to create your first project, so click the *+ Add project* button.

In the dialog that appears, enter *Grocr* as the Project name and select your preferred Country/region:

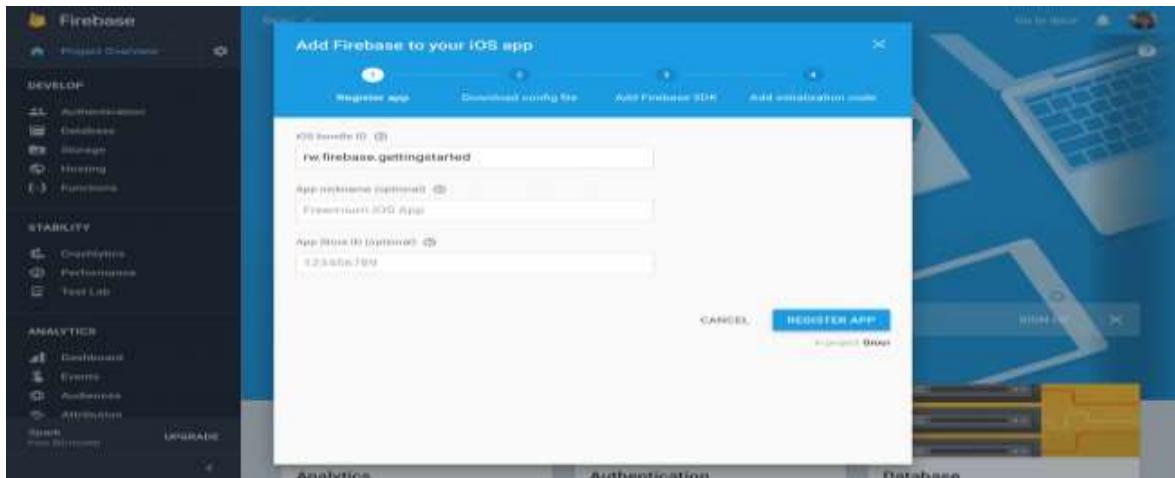


Click ***CREATE PROJECT***, wait for it to create and then press *Continue* to be taken to your project's dashboard:

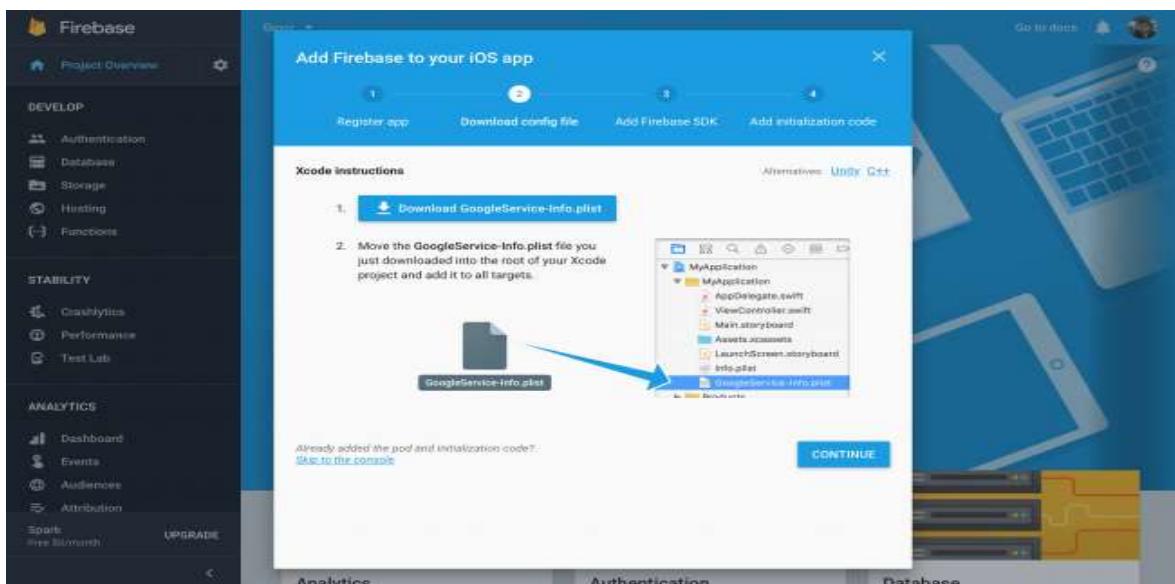


This is a container for your project's Firebase services. You'll use it to store data and authenticate users.

Select the circle iOS button above Add Firebase to your iOS app. Enter `rw.firebaseio.gettingstarted` in the iOS Bundle ID field:

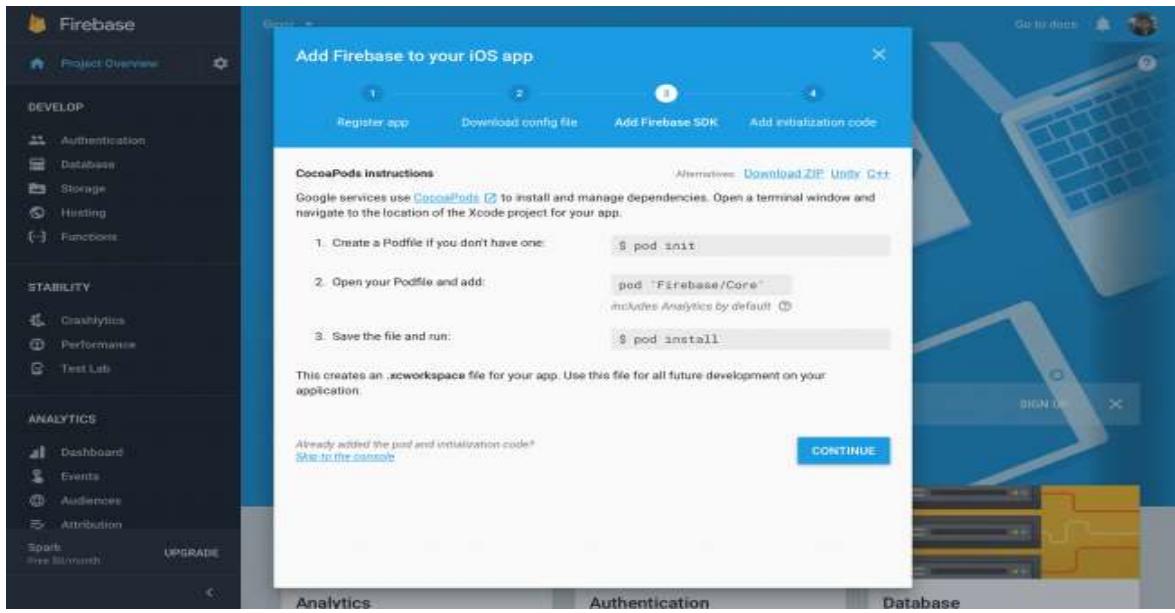


Click REGISTER APP and then click Download GoogleService-Info.plist. Follow the instructions and move it to the Grocr project in XCode.

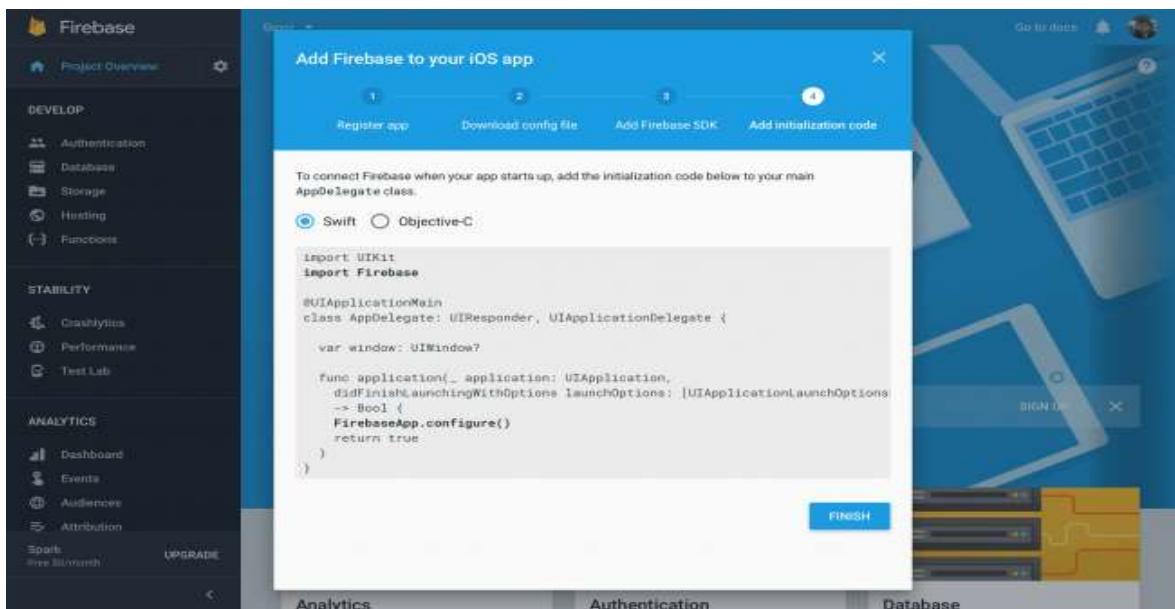


When prompted by XCode, make sure Copy Items if needed is checked.

Return to your Firebase project's webpage and click CONTINUE. The next page describes how to install the Firebase SDK:



The starter project already includes the SDK, so just click CONTINUE. The last page explains how to connect Firebase when your app starts:



You need to do this in the tutorial app. In XCode, open *AppDelegate.swift* and add this code before the return statement within *application(\_:didFinishLaunchingWithOptions:)*:

```
FirebaseApp.configure()
```

Back on your Firebase project's webpage, click FINISH to see your new project's details:

The screenshot shows the Firebase Project Overview page. On the left, there's a sidebar with sections for DEVELOP (Authentication, Database, Storage, Hosting, Functions), STABILITY (Crashlytics, Performance, Test Lab), and ANALYTICS (Dashboard, Events, Audiences, Attribution). A 'Spark' plan is listed as 'Free \$0/month' with an 'UPGRADE' button. The main area is titled 'Overview' with a blue header bar. It shows '1 app in project' and the app name 'rw.firebaseio.gettingstarted'. Below this, there are four metrics: 'Daily active users' (0), 'Monthly active users' (0), 'Crash-free users' (0%), and 'Crashes' (0). At the bottom, there's a 'Discover Firebase' section with three cards: one showing a person at a laptop with an upward arrow, another with a person's profile, and a third with a circuit board.

That's it!

## 2.5 API

An Application Programming Interface (API) is a set of functions, procedures, methods or classes used by computer programs to request services from the operating system, software libraries or any other service providers running on the computer. A computer programmer uses the words in the API to make application programs. Types of API include web services API like the Twitter API, which allows programs to use the API to receive updates on tweets.

APIs can have to do with websites, like the Twitter example, in which programmers use the Twitter API to interact with Twitter, like finding tweets by a certain user or tweeting on a user's behalf. The Facebook API works similarly: it a system that Facebook created that I can use to access certain data on Facebook.

APIs can also be used to interact with systems in general. In the case of the iPhone, there is the language of Objective-C, and the iPhone API is a set of Objective-C methods built by Apple that allow a developer's app to interact with the iPhone. For example, the iPhone API includes methods for responding to fingers on the touch screen, and methods for drawing to the screen.

# **Chapter Three**

# **“Application Design”**

## 3.1 Design Interface

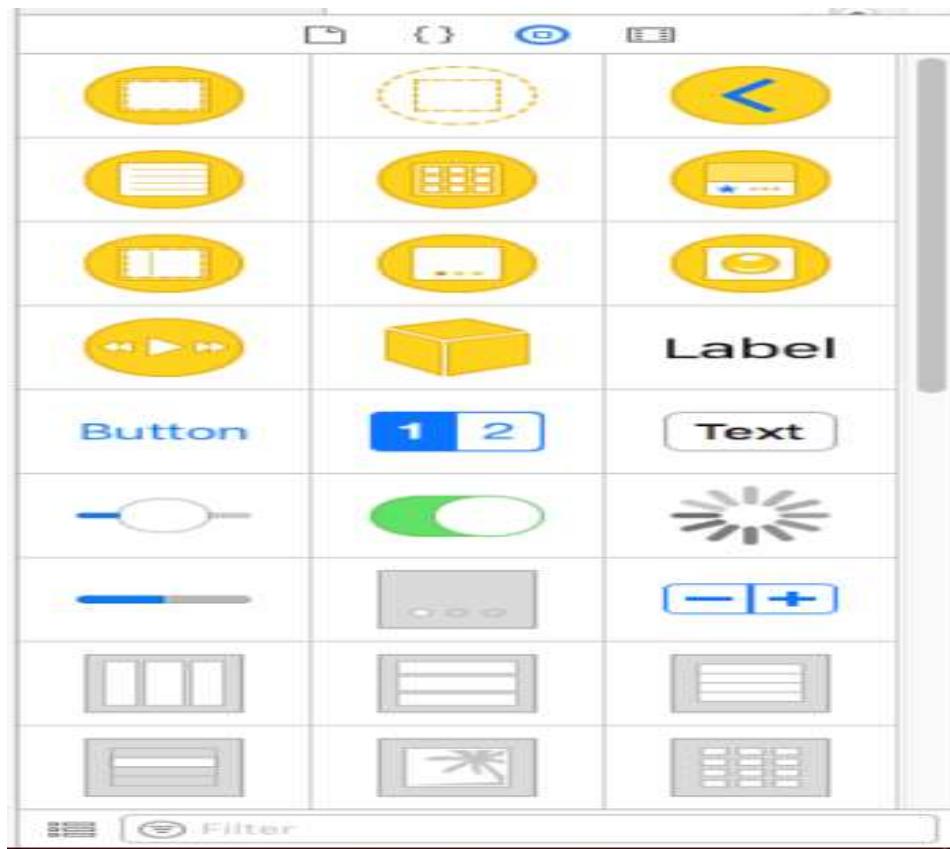
User Interface Design is the design of websites, computers, appliances, machines, mobile communication devices, and software applications with the focus on the user's experience and interaction. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals—what is often called user-centered design.

User interface design is prettying the pixels up. It's a component of user experience design. It's no less important than any other part of the process, and a huge part of what constitutes a great user experience. Many people think this part of design is really the whole shebang, but they're wrong. It makes the experience aesthetically pleasing, but good UI design on poor UX design is still poor design.

User Interface Design is the process of crafting a visual language and hierarchy that allows someone to use and engage an application.

in XCode developer the tools are shown as the following figure.

We can find any tool by searching for it using search bar at bottom of figure.

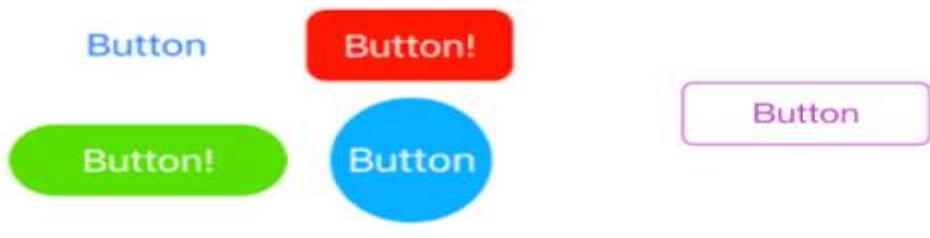


The tools we can use in our application are the following: -

- Button
- Label
- Text Field
- Image
- Search bar
- Table view
- Stare cosmos
- Collection view

### 3.1.1 buttons

Material design buttons allow users to take actions, and make choices, with a single tap. There are many distinct button styles including text buttons, contained buttons, and floating action buttons.



### 3.1.2 labels

A label describes an onscreen interface element or provides a short message. Although people can't edit labels, they can sometimes copy the content of labels. Labels can display any amount of static text but are best kept short.



### 3.1.3 Text Field

A text field is a single-line, fixed-height field, often with rounded corners, that automatically brings up a keyboard when the user taps it. Use a text field to request a small amount of information, such as an email address.



### 3.1.4 Image View

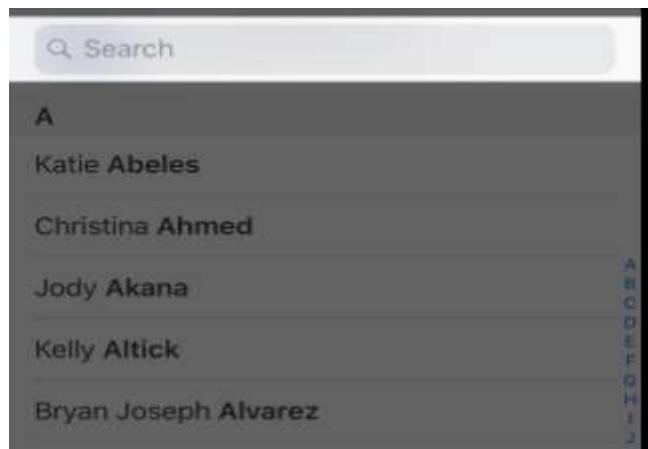
An image view displays a single image or an animated sequence of images over a transparent or opaque background. Within an image view, images may be stretched,

scaled, sized to fit, or pinned to a specific location. Image views are noninteractive by default.



### 3.1.5 Search Bar

A search bar allows people to search through a large collection of values by typing text into a field. A search bar can be displayed alone, or in a navigation bar or content view. When displayed in a navigation bar, a search bar can be pinned to the navigation bar so it's always accessible, or it can be collapsed until the user swipes down to reveal it.



### 3.1.6 Table View

In this lesson, you create a second, table view-based scene, that lists the user's meals. This meal list becomes the initial scene for your app. You also design custom table cells to display each meal.



### 3.1.7 Stare Cosmos

This is a UI control for iOS that shows a 5-star rating or takes rating input from the user. Cosmos is a subclass of a UIView. Let your users see and post beautiful 3-star reviews!

- Shows star rating with an optional text label.
- Can be used as a rating input control.
- Cosmos view can be customized in the Storyboard without writing code.
- Includes different star filling modes: full, half-filled and precise.



### 3.1.8 Collection View

The Collection View provides a flexible way to present content to the user. Similar to a table view, a collection view gets data from custom data source objects and displays it using a combination of cell, layout, and supplementary views. A

collection view can display items in a grid or in a custom layout that you design. In this tutorial we will display a number of colored cells in a collection view.



### 3.2 IFood App Design user login

First page is user login



First the frame has red back color

Second frame contains every of

- Skip button.
- IFood image view.
- User name Text field.
- Password Text field.
- login Button.
- Reset password label with tap gesture action.
- Already member label.
- Sign up label with tap gesture action.

### 3.2.1 Skip button



This button action can skip this page and go to next page.

```
@IBAction func SkipB(_ sender: UIButton) {  
    self.performSegue(withIdentifier: "restaurants",  
                      sender: nil)  
}
```

### 3.2.2 iFood image view.

Select any image and insert it in my image view



### 3.2.3 username and password text field

After entering every of user name and password the value stored in these text fields then waiting for action.

They must be declared in my app as any names As shown in this code

```
@IBOutlet weak var EmailTxtF: TextFieldView!  
@IBOutlet weak var PassTxtF: TextFieldView!
```

And this is the design for text fields.



### 3.2.4 Login button



This button takes every of values entered in every of user name and password text fields and take action if this value stored in my firebase or not If it exists action will turn to next page welcome page If not, it will pop up error message that user is not in my firebase so you should signup.

As shown in the following code.

```
@IBAction func LogInB(_ sender: UIButton) {
    Auth.auth().signIn(withEmail: EmailTxtF.text!, password: PassTxtF.text!) { (User, Error) in
        if Error == nil {
            print("You are IN")
            self.performSegue(withIdentifier: "resturants", sender: nil)
        } else {
            MessageBox.show(mesg: Error!.localizedDescription, MyVC: self)
        }
    }
}
```

### 3.2.5 Reset password label with tap gesture action



This label takes an action called tap gesture action that turn this label as a button

Its worked as user tap on it will open reset password page as in following code.

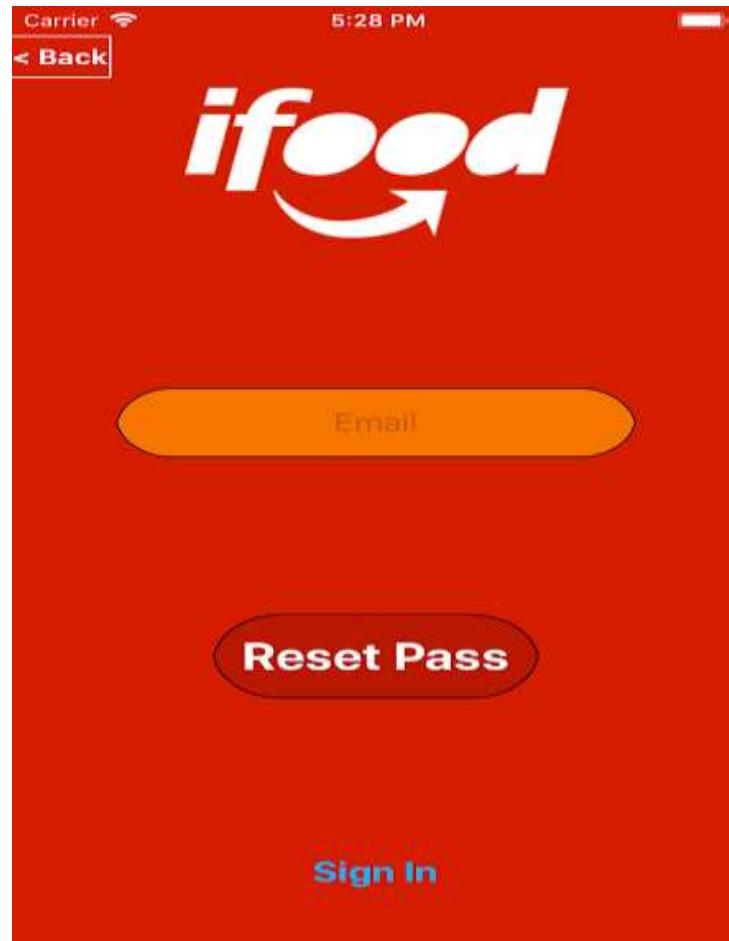
```
@IBAction func ResetPassTapGesture(_ sender: Any) {
    performSegue(withIdentifier: "resetPassSegue",
    sender: nil)
}
```

And also, the same in Sign up label with tap gesture action



### 3.3 Reset Password

second page is Reset Password.



First the frame has red back color.

Second frame contains every of

- Skip button.
- IFood image view.
- Email Text field.
- Reset pass Button.
- Sign in label with tap gesture action.

### 3.3.1 Skip button



This button action can skip this page and go to next page.

```
@IBAction func BackB(_ sender: UIButton) {  
    self.dismiss(animated: true, completion: nil)  
}
```

### 3.3.2 iFood image view.

Select any image and insert it in my image view



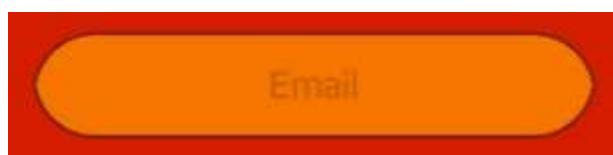
### 3.3.3 Email Text field

After entering every of user email the value stored in these text fields then waiting for action.

They must be declared in my app as any names as shown in this code

```
@IBOutlet weak var ResetPassTF: TextFieldView!
```

And this is the design for text fields.



### 3.3.4 Reset pass Button



This button takes every of values entered in email text fields and take action if this value stored in my firebase or.

As shown in the following code.

```
@IBAction func ResetPassB(_ sender: ButtonView) {
    Auth.auth().sendPasswordReset(withEmail:
        ResetPassTF.text!) { (Error) in
            if Error == nil{
                MessageBox.show(mesg: "We send your new
                    password into you Email", MyVC: self)
            }else{
                MessageBox.show(mesg: Error.
                    debugDescription, MyVC: self)
            }
        }
}
```

### 3.3.5 Sign in label with tap gesture action



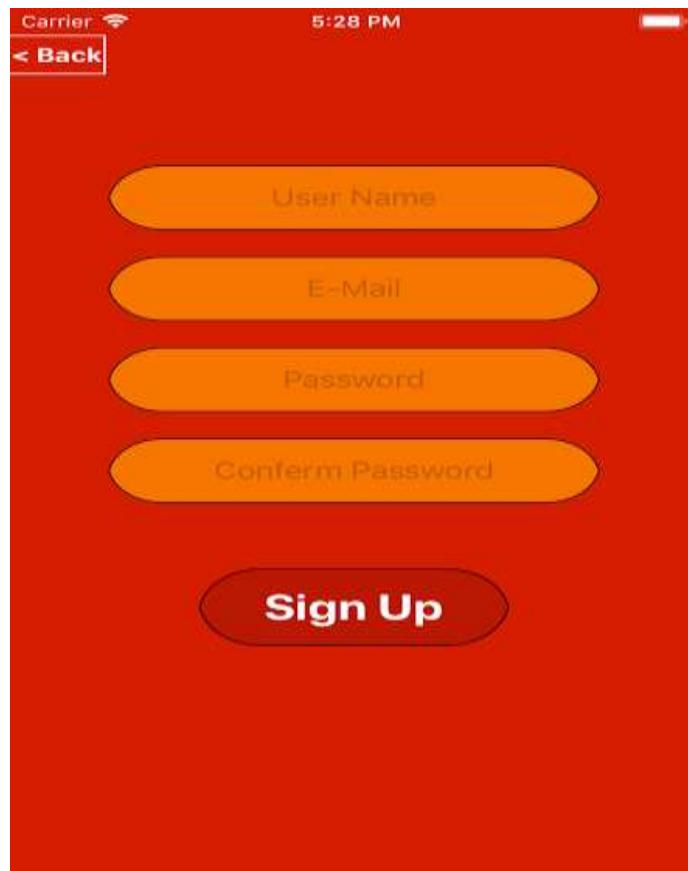
This label takes an action called tap gesture action that turn this label as a button

Its worked as user tap on it will open sign in page as in following code.

```
@IBAction func SignInTapGesture(_ sender: Any) {
    self.dismiss(animated: true, completion: nil)
}
```

### **3.4 sign up**

Third page is Sign Up.



First the frame has red back color.

Second frame contains every of

- Skip button.
- User Name Text field.
- Email Text field.
- Password Text field.
- Confirm Password Text field.
- Sign Up Button.

### 3.4.1 Skip button



This button action can skip this page and go to previous page.

```
@IBAction func BackB(_ sender: UIButton) {  
    self.dismiss(animated: true, completion: nil)  
}
```

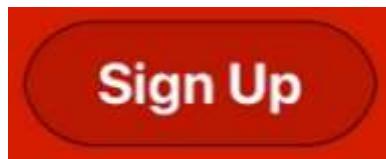
### 3.4.2 User Name, Email, Password and Confirm Password Text fields.



After entering every of user name, email, password and confirm password the value stored in these text fields then waiting for action. They must be declared in my app as any names as shown in this code

```
@IBOutlet weak var NameTextF: UITextField!  
@IBOutlet weak var EmailTextF: TextFieldView!  
@IBOutlet weak var PasswordTextF: TextFieldView!  
@IBOutlet weak var CPasswordTextF: TextFieldView!
```

### 3.4.3 Sign Up Button



This button takes every of values entered in user name, email, password and confirm password text fields and take action then store data of new user in my fire base.

As shown in the following code.

```
@IBAction func SignUp(_ sender: UIButton) {  
    //SignUp into fire base with email and password  
    if (NameTextF.text != "")&&(PasswordTextF.text != "")&&(CPasswordTextF.text != ""){  
        if PasswordTextF.text! == CPasswordTextF.text!{  
            Auth.auth().createUser(withEmail:  
                EmailTextF.text!, password:  
                PasswordTextF.text!) { (User, Error) in  
  
                if Error == nil{  
                    print("Done")  
                    try! Auth.auth().signOut()  
                    self.dismiss(animated: true,  
                        completion: nil)  
                }  
  
                }else{  
                    MessageBox.show(mesg: Error!.localizedDescription, MyVC:  
                        self)  
                }  
            }  
            }else{  
                MessageBox.show(mesg: "Password Not Equal  
                Confirm Password", MyVC: self)  
            }  
            }else{  
                MessageBox.show(mesg: "Fill Required Text",  
                MyVC: self)  
            }  
    }  
}
```

### 3.4 Home Page

Fourth page is Restaurant page.



frame contains every of

- Search Bar.
- Table View.

### 3.4.1 Search Bar



It is very important in any application for searching any thing you need in application So we can manage this search bar to find any name of restaurant that user need as following code.

```
func searchBar(_ searchBar: UISearchBar, textDidChange searchText: String) {
    if searchText == "" {
        parseData(){}
    }else{
        restaurants = restaurants.filter({ (name) -> Bool
            in
            return name.name.lowercased().contains
                (searchText.lowercased())
        })
    }
    self.tableView.reloadData()
}
```

### 3.4.2 Table View

This is a declaration of table view and each of label of restaurant name and image view of restaurant image and cosmos stare for restaurant rating.

```

class TableCellView: UITableViewCell {
    @IBOutlet weak var cosmosView: CosmosView!
    @IBOutlet weak var resturantName: UILabel!
    @IBOutlet weak var ImageView: UIImageView!

    override func awakeFromNib() {
        super.awakeFromNib()

        self.layer.borderWidth = 1
        self.layer.borderColor = UIColor.black.
            withAlphaComponent(0.5).cgColor
        self.layer.shadowRadius = 3
        self.layer.shadowColor = UIColor.black.cgColor
        self.layer.shadowOffset = CGSize(width: 2, height:
            2)
        self.layer.shadowOpacity = 0.8
        self.clipsToBounds = true
    }
}

```



As shown in previous image the table view consists of cells and each cell contains every of

- Opacity View.
- Back Ground Image.

### 3.4.2.1 Opacity View

The opacity view is a tool with no background or color it just transparent

And contains many tools then show them.



This opacity view contains

- Label: show restaurant name.
- Cosmos stare: to manage rating for restaurants.

```
func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath) as! TableViewCell
    cell.restaurantName.text =
        restaurants[indexPath.row].name.capitalized
    cell.cosmosView.rating = Double(restaurants[indexPath.row].rate)
    cell.ImageView.setCustomImage(restaurants[indexPath.row].img, contentMode: .scaleToFill, scale: 0.5)
    return cell
}
```

```
func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    performSegue(withIdentifier: "eachRestaurantSegue",
                sender: self)
}

override func prepare(for segue: UIStoryboardSegue,
                      sender: Any?) {
    if let destination = segue.destination as?
        EachRestaurantVC {
        destination.restaurant = restaurants[(TableView.
            indexPathForSelectedRow?.row)!]
    }
}
```

But we first can call the application restaurant from my Json URL

```
override func viewDidLoad() {
    super.viewDidLoad()

    TableView.tableFooterView = UIView()
    TableView.separatorInset = .zero
    TableView.contentInset = .zero
    TableView.delegate = self
    TableView.dataSource = self

    parseData()
    searchBar()
}

func parseData(completed: @escaping () -> ()) {
    guard let url = URL(string: "http://my-json-server.typicode.com/erafa93/IFood/restaurants")
    else { return }
    URLSession.shared.dataTask(with: url) { (data, response, error) in
        if error == nil {
            do {
                self.restaurants = try JSONDecoder().decode([Rests].self, from: data!)
                DispatchQueue.main.async {
                    completed()
                    self.tableView.reloadData()
                }
            } catch { print("Error", error) }
        }
    }.resume()
}
```

### 3.4.2.2 Back Ground Image

Insert restaurant image by calling it from my server as showed in previous code.



the image behind the opacity view.

### 3.5 Restaurant Page

fifth page is Restaurant Information Page.



First the frame has red back color.

Second frame contains every of

- Skip button.
- Restaurant Image View.
- Restaurant name Label.
- Restaurant Location Label.
- Restaurant phone Label.
- Restaurant Rate Cosmos Stare.
- Restaurant Food button.

### 3.5.1 Skip button



This button action can skip this page and go to previous page.

```
@IBAction func BackB(_ sender: UIButton) {  
    self.dismiss(animated: true, completion: nil)  
}
```

### 3.5.2 Restaurant Image View, Restaurant name Label, Restaurant Location Label, Restaurant phone Label, Restaurant Rate Cosmos Stare.



Append each information about all restaurants from my Json URL.

<http://my-json-server.typicode.com/arafa93/lFood/resturants>

by the following code.

```

import UIKit
import Cosmos

class EachRestaurantVC: UIViewController {

    @IBOutlet weak var cosmosView: CosmosView!
    var restaurant:Rests?
    var food = [Food]()

    @IBOutlet weak var restaurantPhone: UILabel!
    @IBOutlet weak var restaurantLocation: UILabel!
    @IBOutlet weak var restaurantName: UILabel!
    @IBOutlet weak var imageView: UIImageView!

    @IBAction func BackB(_ sender: UIButton) {
        self.dismiss(animated: true, completion: nil)
    }

    override var preferredStatusBarStyle : UIStatusBarStyle {
        return .lightContent
    }
    override func viewDidLoad() {
        super.viewDidLoad()
        restaurantName.text = restaurant?.name
        restaurantLocation.text = restaurant?.location
        restaurantPhone.text = restaurant?.phone
        cosmosView.rating = Double((restaurant?.rate)!)
        imageView.setCustomImage(restaurant?.img,
            contentMode: .scaleToFill, scale: 0.5)
    }
}

```

### 3.5.3 Restaurant Food button.



This button allow user to move to restaurant foods page and show all foods.

```

@IBAction func allFoodB(_ sender: ButtonView) {
    performSegue(withIdentifier: "allFood", sender:
    self)
}

```

### 3.6 Food page

Sixth page is Restaurant Food page.



First the frame has red back color.

Second frame contains every of

- Skip Button.
- Collection View.

### 3.6.1 Skip button



This button action can skip this page and go to previous page.

```
@IBAction func BackB(_ sender: UIButton) {
    self.dismiss(animated: true, completion: nil)
}
```

### 3.6.2 Collection View

This is a declaration of collection view consist of cell each of cell contains every of label of food name and label of food description and label of food price and image view of food image and cosmos stare for food rating.

```
import UIKit
import Cosmos
class CollectionViewCell: UICollectionViewCell {

    @IBOutlet weak var Image: UIImageView!
    @IBOutlet weak var nameLB: UILabel!
    @IBOutlet weak var contentLB: UILabel!
    @IBOutlet weak var priceLB: UILabel!
    @IBOutlet weak var cosmosView: CosmosView!

    override func awakeFromNib() {
        super.awakeFromNib()

        self.layer.borderWidth = 1
        self.layer.borderColor = UIColor.black.
            withAlphaComponent(0.5).cgColor
        self.layer.shadowRadius = 3
        self.layer.shadowColor = UIColor.black.cgColor
        self.layer.shadowOffset = CGSize(width: 2, height:
            2)
        self.layer.shadowOpacity = 0.8

        self.clipsToBounds = true
    }
}
```



As shown in previous image the collection view consists of cells and each cell contains every of

- Image View.
- Food Name Label.
- Food description Label.
- Food Price Label.
- Cosmos Rating Stare.

### 3.6.2.1 Image View and food (name, Description, price and rating).

But we first can call the application food information from my Json URL to be appended in food page.

```

import UIKit

class AllFoodVC: UIViewController , UICollectionViewDelegate , UICollectionViewDataSource {
    var food = [[String:Any]]()
    var name = [String]()
    var content = [String]()
    var salary = [String]()
    var rate = [Int]()
    var image = [String]()
    var RestaurantNumber = Int()

    override var preferredStatusBarStyle : UIStatusBarStyle {
        return .lightContent
    }

    @IBOutlet weak var CollectionV: UICollectionView!

    override func viewDidLoad() {
        super.viewDidLoad()

        CollectionV.delegate=self
        CollectionV.dataSource=self

        parsURL(){}
    }

    let strPhoneNumber = "01201173768"
    @IBAction func BackB(_ sender: UIButton) {
        self.dismiss(animated: true, completion: nil)
    }

    func collectionView(_ collectionView: UICollectionView,
                       numberOfItemsInSection section: Int) -> Int {
        return name.count
    }
    func collectionView(_ collectionView: UICollectionView,
                       cellForItemAt indexPath: IndexPath) ->
        UICollectionViewCell {
        let cell = CollectionV.dequeueReusableCell(withIdentifier: "CVC", for: indexPath)
        as! CollectionViewCell
        cell.nameLB.text = name[indexPath.row]
        cell.contentLB.text = content[indexPath.row]
        cell.priceLB.text = salary[indexPath.row]
        cell.cosmosView.rating = Double(rate[indexPath.row])
        cell.Image.setCustomImage(image[indexPath.row],
                                  contentMode: .scaleToFill, scale: 0.5)
        return cell
    }
    func parsURL(completed : @escaping ()->()){
        let url = URL(string: "http://my-json-server.typicode.com/arafa93/IFood/resturants/\(RestaurantNumber)")
        URLSession.shared.dataTask(with: url!){ (data,
            response,error) in
            if error != nil{
                print("didn't work,\(String(describing: error))")
            }
            else{
                do{
                    let parseData = try JSONSerialization.jsonObject(with: data!, options:.
                        mutableLeaves) as! [String:Any]
                    DispatchQueue.main.async {
                        completed()
                        self.CollectionV.reloadData()
                    }
                }
            }
        }
    }
}

```

Notes

```
[1]
for (key,value) in parseData{
    if (key == "foods"){
        if let foodsA:=[[String:Any]] = value as? [[String:Any]]{
            //self.food = foodsA

            for dic in foodsA{
                for (key,value) in dic{

                    if key == "name"{
                        self.name.
                        append(value as!
String)
                    }else if key ==
"content"{
                        self.content.
                        append(value as!
String)
                    }else if key ==
"rate"{
                        self.rate.
                        append(value as!
Int)
                    }else if key ==
"image"{
                        self.image.
                        append(value as!
String)
                    }else if key ==
"salary"{
                        self.salary.
                        append(value as!
String)}}}}}}}
catch let error as NSError{print(error)}}}}.
resume()}
```

# **Chapter Four "Sign In Validation Using FireBase"**

In this chapter we can mange the ability to store user's signup information to allow sign in approval.

Before the installation of firebase, it cannot have applied before install and download cocoaPods.

## 4.1 What is cocoaPods

CocoaPods is a dependency manager for Swift and Objective-C Cocoa projects. It has over 48 thousand libraries and is used in over 3 million apps. CocoaPods can help you scale your projects elegantly.

### 4.1.1 install cocoaPods

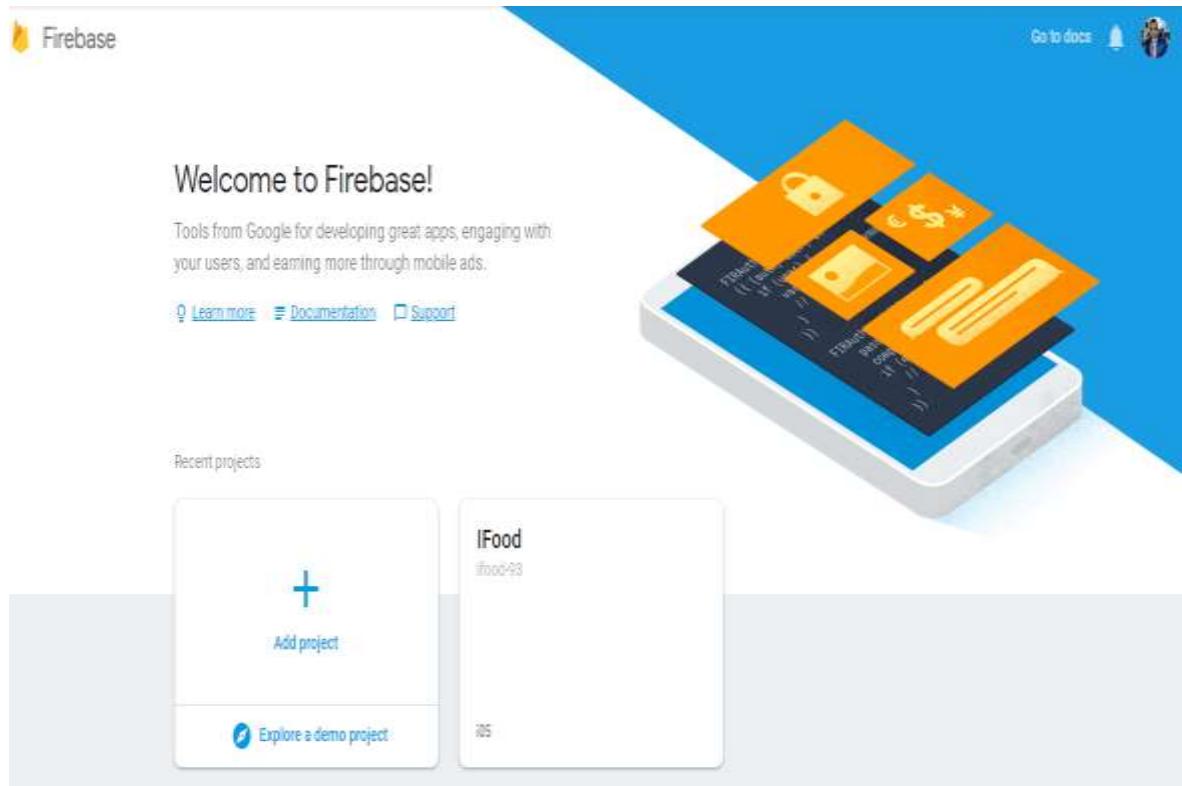
CocoaPods is built with Ruby and is installable with the default Ruby available on OS X. We recommend you use the default ruby.

Using the default Ruby install can require you to use **sudo** when installing gems. Further installation instructions are in [the guides](#).

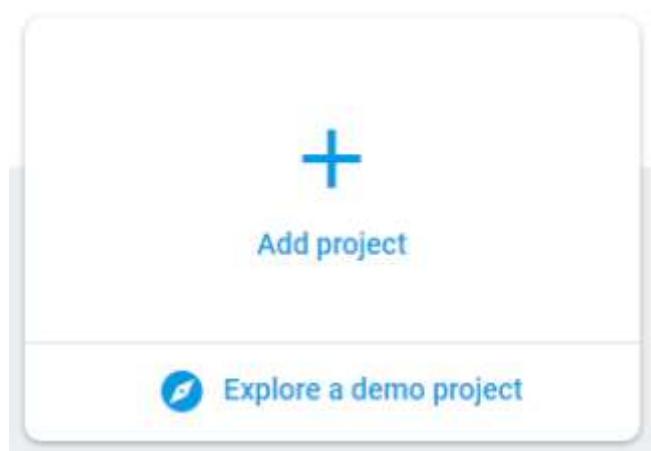
```
# Xcode 8 + 9  
$ sudo gem install cocoapods
```

## 4.1.2 connect App to Firebase

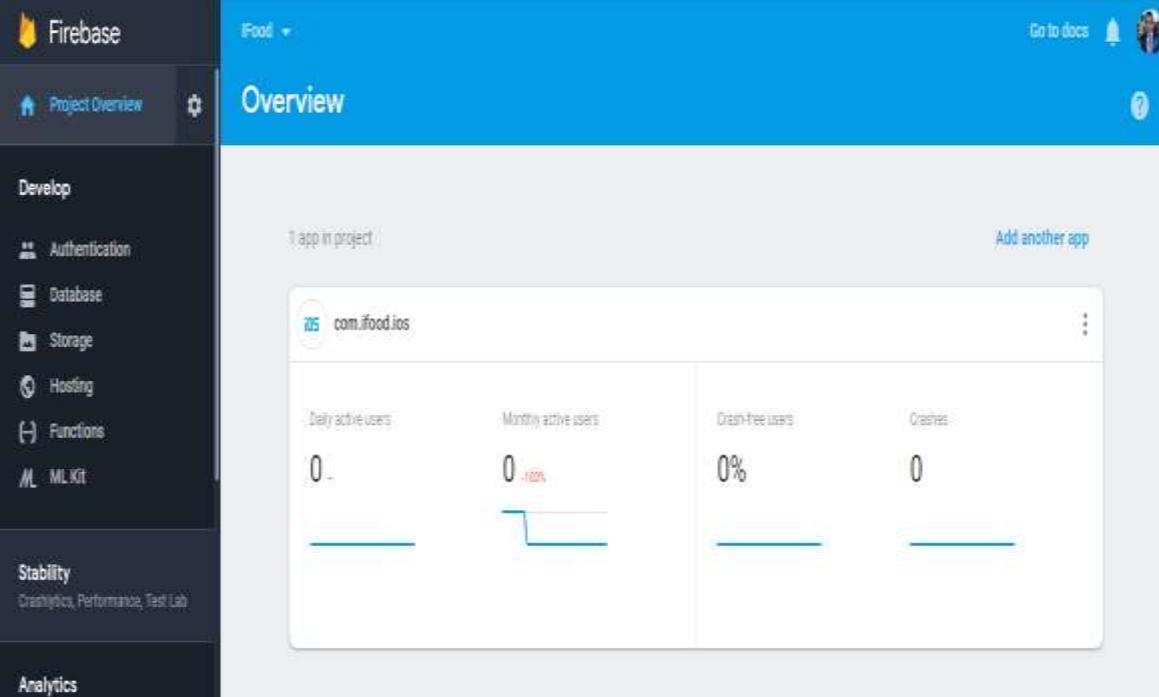
First open firebase home page



Second choose add project option to upload IFood application

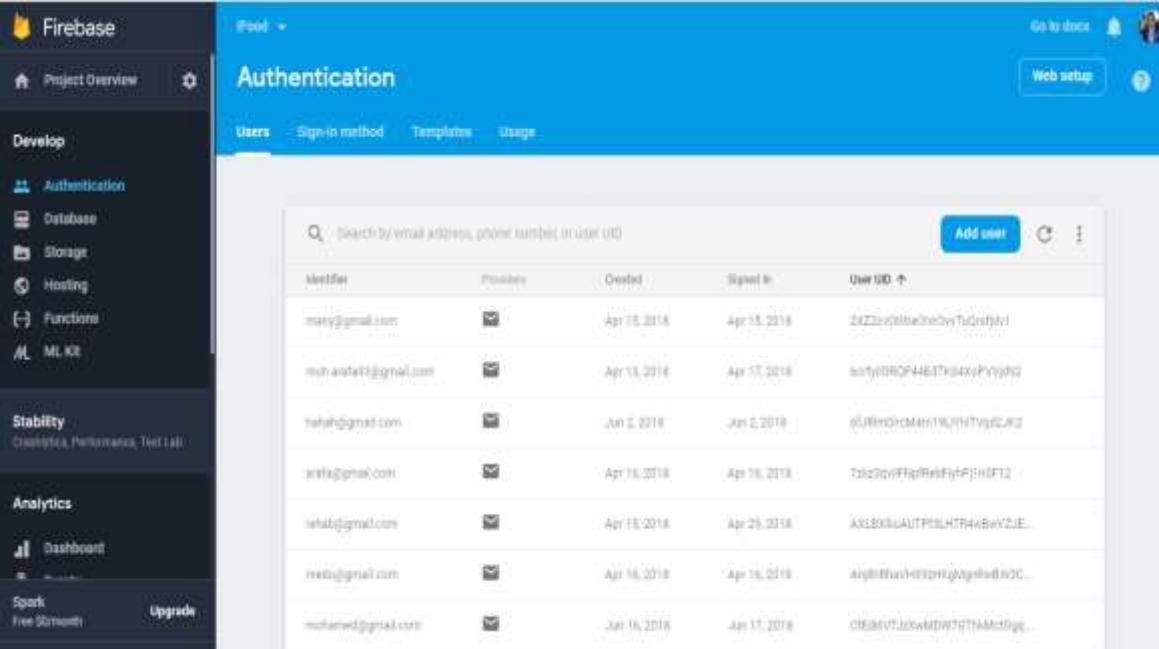


Third after upload application it must open the app database and options used in it.



The screenshot shows the Firebase Project Overview page. On the left, there's a sidebar with 'Project Overview' at the top, followed by 'Develop' and several services: Authentication, Database, Storage, Hosting, Functions, and ML Kit. Below that is 'Stability' and 'Analytics'. The main area has a blue header 'Overview' with a dropdown arrow. It shows '1 app in project' and a button 'Add another app'. A card for 'com.food.ios' is displayed with metrics: Daily active users (0), Monthly active users (0), Crash-free users (0%), and Crashes (0). There are also line graphs for each metric over time.

Here the app database and how to manipulate accounts from authentication option.



The screenshot shows the Firebase Authentication screen. The sidebar includes 'Authentication' under 'Develop' and other services like Database, Storage, and Hosting. The main area has tabs for 'Users', 'Sign-in method', 'Templates', and 'Usage'. A search bar at the top says 'Q (search by email address, phone number, or user ID)'. A 'Web setup' button is in the top right. Below is a table of user accounts:

Identifier	Provider	Created	Signed In	User ID
mary@gmail.com	Email	Apr 15, 2018	Apr 15, 2018	2AZ2zv5lmttce5yTu2rljtlv1
moh.aftab2@gmail.com	Email	Apr 13, 2018	Apr 17, 2018	b0f10DOP44B1TK54xjPyvjd0
hahah@gmail.com	Email	Jun 2, 2018	Jan 2, 2019	6JURh9hMAH116jHTVq2AFZ
arsh@gmail.com	Email	Apr 16, 2018	Apr 16, 2018	TzLzRqvFRpRsfPfPjhsFT12
rehab@gmail.com	Email	Apr 18, 2018	Apr 21, 2018	AxLB03sAUTP03LHTR46BwvZJE...
rehab@gmail.com	Email	Apr 16, 2018	Apr 16, 2018	AxLB03sAUTP03LHTR46BwvZC...
mohamed@gmail.com	Email	Jul 16, 2018	Apr 17, 2018	ClearV7JxWMDn7GTHMe7fg...

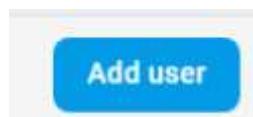
It shows all information of users and restaurants accounts.

Such as: -

- Identifier: login user name.
- Provider: the authority for account.
- Created: account date when created.
- Sign in: all dates when user sign in.
- User UID: by default, id created by firebase.

Identifier	Provider	Created	Signed in	User UID ↗
mamym@gmail.com	✉	Apr 15, 2018	Apr 15, 2018	EXZ2cvOotieOvrOvvTuQsfjduI
moh.matew93@gmail.com	✉	Apr 15, 2018	Apr 17, 2018	6crfyIGRQP4AEEdTRSAeoP
hahah@gmail.com	✉	Jun 2, 2018	Jun 2, 2018	6fJRLmGrcM4mT9LYnITVgd2-JK2
arafa@gmail.com	✉	Apr 16, 2018	Apr 16, 2018	7zklzIqvIPNpfRebFyhPj1nSF12
rehab@gmail.com	✉	Apr 15, 2018	Apr 29, 2018	AXLBX0iuAUTP53LHTR4wBwVZIE...

And has ability to add any user directly from firebase without using app sign up.



And more can do all of modification options as

Reset Password.

Disable Account.

Delete Account.

Search by email address, phone number, or user ID					Add user	⋮
Identifier	Provider	Created	Signed in	User UID ↗		
mamym@gmail.com	✉	Apr 15, 2018	Apr 15, 2018	EXZ2cvOotieOvrOvvTuQsfjduI	Reset password	
moh.matew93@gmail.com	✉	Apr 15, 2018	Apr 17, 2018	6crfyIGRQP4AEEdTRSAeoP	Disable account	
hahah@gmail.com	✉	Jun 2, 2018	Jun 2, 2018	6fJRLmGrcM4mT9LYnITVgd2-JK2	Delete account	
arafa@gmail.com	✉	Apr 16, 2018	Apr 16, 2018	7zklzIqvIPNpfRebFyhPj1nSF12		

That's all of how to work in firebase then how to call this database in IOS Xcode App.

First import firebase in every of sign in, sign up and reset password page code.swift



```
import FirebaseAuth
```

In sign in must call function from firebase library pods to manage sign in confirmation is

Auth.auth().signin()

```
@IBAction func LogInB(_ sender: UIButton) {
    Auth.auth().signIn(withEmail: EmailTxtF.text!, password: PassTxtF.text!) { (User, Error) in
        if Error == nil {
            print("You are IN")
            self.performSegue(withIdentifier: "resturants", sender: nil)
        }else{
            MessageBox.show(mesg: Error!.localizedDescription, MyVC: self)
        }
    }
}
```

In sign up must call function from firebase library pods to manage sign in confirmation is

Auth.auth().CreateUser()

```
 @IBAction func Signup(_ sender: UIButton) {
    //Submit into fire base with email and password
    if (NameTxtF.text != "")&(PasswordTxtF.text != "")&(ConfirmTxtF.text != ""){
        if PasswordTxtF.text == ConfirmTxtF.text{
            Auth.auth().createUser(withEmail: EmailTxtF.text!, password: PasswordTxtF.text!) { (User, Error) in
                if Error == nil{
                    print("Done")
                    self.Auth.auth().signOut()
                    self.dismiss(animated: true, completion: nil)

                }else{
                    MessageBox.show(mesg: Error!.localizedDescription, MyVC: self)
                }
            }
        }else{
            MessageBox.show(mesg: "Password Not Equal Confirm Password", MyVC: self)
        }
    }else{
        MessageBox.show(mesg: "Email Required Text", MyVC: self)
    }
}
```

In reset password must call function from firebase library pods to manage sign in confirmation is

Auth.auth().SendPasswordReset()

```
@IBAction func ResetPassB(_ sender: ButtonView) {
    Auth.auth().sendPasswordReset(withEmail: ResetPassTF.text!) { (Error) in
        if Error == nil{
            MessageBox.show(mesg: "We send your new password into you Email", MyVC: self)
        }else{
            MessageBox.show(mesg: Error.debugDescription, MyVC: self)
        }
    }
}

@IBAction func SignInTapGesture(_ sender: Any) {
    self.dismiss(animated: true, completion: nil)
}
```

At now all things is right by create database for application in firebase then connect firebase to my code.

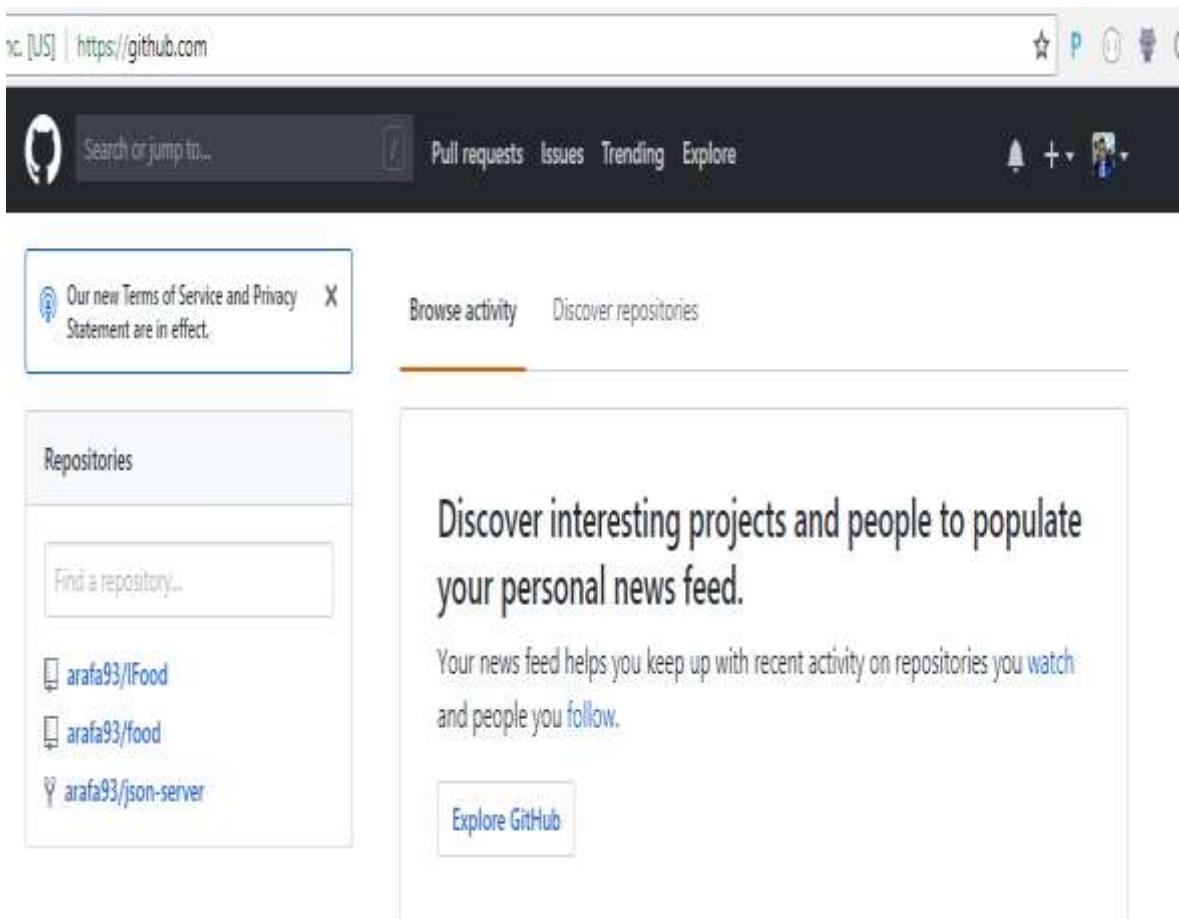
# **Chapter five “connect Application to server”**

## 5.1 GitHub

As we explain what GitHub in is chapter two.

GitHub is a website and service that we hear geeks rave about all the time, yet a lot of people don't really understand what it does. Want to know what all the GitHub hubbub is about? Read on to find out.

We managed our json code in GitHub to apply URL for calling it to my server.



[arafa93 / IFood](#)

Code Issues Pull requests Wiki Releases More Settings

No description, website, or topics provided.

Add topics Edit

82 commits 1 branch 0 releases 1 contributor

Branch: master ▾ Create new file Find file Clone or download ▾

 arafa93 Update db.json Latest commit 65b663e on May 13

 .gitignore initial commit 2 months ago

 db.json Update db.json 2 months ago

Help people interested in this repository understand your project by adding a README.

Add a README

 Search or jump to... Pull requests Issues Trending Explore

[arafa93 / IFood](#)

Code Issues Pull requests Wiki Releases More Settings

Branch: master ▾ IFood / db.json Find file Copy path

 arafa93 Update db.json 65b663e on May 13

1 contributor

139 lines (138 sloc) 5.23 KB

Copy Raw Blame History

```
1  {
2   "restaurants": [
3     [
4       "id": 1,
5       "name": "KFC - Zagazig",
6       "img": "https://europe-public.foodpanda.com/dynamic/production/eg/images/vendors/e5bi_sap.jpg?v=20150909144409",
7       "location": "1 Talaat Harb, Mosque, Qesm Than AZ Zagazig, Ash Sharqia Governorate",
8       "phone": "01281173768",
9       "rate": 4,
10      "Foods": [
11        [
12          "id": 1,
13          "name": "Chicken Meal ( Menu 25 )",
14          "content": "Chicken piece, small plain rice, small French fries and bun ",
15          "Restaurant": "KFC - Zagazig"
16        ]
17      ]
18    ]
19  ]
```

**And json code is: -**

```
{  
  "resturants": [  
    {  
      "id": 1,  
      "name": "KFC - Zagazig",  
      "img": "https://europe-public.foodpanda.com/dynamic/production/eg/images/vendors/e5bi_sqp.jpg?v=20150909144409",  
      "location": "1 Talaat Harb, Mosque, Qesm Than AZ Zagazig, Ash Sharqia Governorate",  
      "phone": "01201173768",  
      "rate": 4,  
      "foods": [  
        {  
          "id": 1,  
          "name": "Chicken Meal (Menu 25)",  
          "content": "Chicken piece, small plain rice, small French fries and bun ",  
          "salary": "EGY 21.93",  
          "image": "https://4.bp.blogspot.com/-dJeKqjD8J0w/UYPMuifGrl/AAAAAAAAXc/ti_j4JYVGfU/s800/raising-canapes-chicken-fingers-box-03.JPG",  
          "rate": 5  
        },  
        {  
          "id": 1,  
          "name": "Chicken Meal (Menu 25)",  
          "content": "Chicken piece, small plain rice, small French fries and bun ",  
          "salary": "EGY 21.93",  
        }  
      ]  
    }  
  ]  
}
```

```
        "image": "https://4.bp.blogspot.com/-dJeKqjD8J0w/UYPMuifGrl/AAAAAAAAXc/ti_j4JYVGfU/s800/raising-canesthicken-fingers-box-03.JPG",
        "rate": 5
    }
]
},
{
    "id": 2,
    "name": "McDonald's - Zagazig",
    "img": "https://europe-public.foodpanda.com/dynamic/production/eg/images/vendors/e0ut_sqp.png?v=20161109104808",
    "location": "Tolba Awaida St, Shaibet an Nakareyah, Markaz El-Zakazik, Ash Sharqia Governorate",
    "phone": "01201173768",
    "rate": 3,
    "foods": [
        {
            "id": 1,
            "name": "Chicken Rice with Sauce Meal",
            "content": "2 Pieces fried chicken, rice and chili sauce or barbeque sauce",
            "salary": "EGY54.39",
            "image": "https://4.bp.blogspot.com/-dJeKqjD8J0w/UYPMuifGrl/AAAAAAAAXc/ti_j4JYVGfU/s800/raising-canesthicken-fingers-box-03.JPG",
            "rate": 5
        }
    ]
},
```

```
{  
    "id": 3,  
    "name": "Cook Door - Zagazig",  
    "img": "https://europe-public.foodpanda.com/dynamic/production/eg/images/vendors/e2hq_sqp.jpg?v=20150913104715",  
    "location": "El-Hariry, Qesm Awal AZ Zagazig, Ash Sharqia Governorate",  
    "phone": "01201173768",  
    "rate": 5,  
    "foods": [  
        {  
            "id": 1,  
            "name": "Large Plain Rice Offer ",  
            "content": "Served with Pepsi can",  
            "salary": "EGY15.35 ",  
            "image": "https://4.bp.blogspot.com/-dJeKqjD8J0w/UYPMuifGrl/AAAAAAAAXc/ti_j4JYVGfU/s800/raising-canapes-chicken-fingers-box-03.JPG",  
            "rate": 5  
        }  
    ],  
    {  
        "id": 4,  
        "name": "Pizza Hut - Zagazig",  
        "img": "https://europe-public.foodpanda.com/dynamic/production/eg/images/vendors/e8xv_sqp.jpg?v=20150406145607",  
        "location": "Al Hasan Ben Al Haitham, Shaibet an Nakareyah, Markaz El-Zakazik, Ash Sharqia Governorate",  
    },  
}
```

```
"phone": "01201173768",
"rate": 5,
"foods": [
{
"id": 1,
"name": "Super Meal",
"content": "2 Pieces fried chicken, served with 200 gm large plain rice, small french fries, bun bread and soft drink",
"salary": "EGY 41.50",
"image": "https://4.bp.blogspot.com/-dJeKqjD8J0w/UYPMuiifGrl/AAAAAAAAXc/ti_j4JYVGfU/s800/raising-canес-chicken-fingers-box-03.JPG",
"rate": 5
},
],
},
{
"id": 5,
"name": "Mo'men - Zagazig",
"img": "https://europe-public.foodpanda.com/dynamic/production/eg/images/vendors/e8yp_sqp.png",
"location": "Tolba Awaida St, Shaibet an Nakareyah, Markaz AZ Zagazig, Ash Sharqia Governorate",
"phone": "01201173768",
"rate": 5,
"foods": [
{
"id": 1,
"name": "Snack Box",
```

```
"content": "2 Pieces fried chicken, served with small French fries and bun bread",  
"salary": " EGY 31.14",  
"image": "https://4.bp.blogspot.com/-dJeKqjD8J0w/UYPMuifGrI/AAAAAAAAXc/ti_j4JYVGfU/s800/raising-caneshicken-fingers-box-03.JPG",  
"rate": 5  
}  
]  
,  
{  
"id": 6,  
"name": "BBQ - Hay El Zohour",  
"img": "https://europe-public.foodpanda.com/dynamic/production/eg/images/vendors/e0nx_sqp.jpg?v=20180306114357",  
"location": "Qawmia, Al-Gawish",  
"phone": "01201173768",  
"rate": 4,  
"foods": [  
{  
"id": 1,  
"name": "Dinner Box",  
"content": "3 Pieces fried chicken, served with small French fries, coleslaw salad and bun bread",  
"salary": "EGY 54.39",  
"image": "https://4.bp.blogspot.com/-dJeKqjD8J0w/UYPMuifGrI/AAAAAAAAXc/ti_j4JYVGfU/s800/raising-caneshichen-fingers-box-03.JPG",  
"rate": 5  
}
```

```
        ]  
    },  
    {  
        "id": 7,  
        "name": "Al Batal Crepe - Sharea' El Farouq",  
        "img": "https://europe-public.foodpanda.com/dynamic/production/eg/images/vendors/e2nf_sqp.jpg?v=20180301132911",  
        "location": "171 E Broadway, New York, NY 10002",  
        "phone": "01201173768",  
        "rate": 5,  
        "foods": [  
            {  
                "id": 1,  
                "name": "Mix 2 Meal – Small",  
                "content": "Pick 2 sandwiches from Cheeseburger, Chicken MacDo® and Beef Burger + Small Fries + Regular Drink",  
                "salary": "EGY 36.84",  
                "image": "https://4.bp.blogspot.com/-dJeKqjD8J0w/UYPMuiifGrI/AAAAAAAAXc/ti_j4JYVGfU/s800/raising-canes-chicken-fingers-box-03.JPG",  
                "rate": 5  
            }  
        ]  
    }  
]
```

## 5.2 Json Server

It is a server that provide some free services to users we mange allowing to use our application json code in it for free but offer few services.

How to create it?

By doing the following steps

The screenshot shows the homepage of <https://my-json-server.typicode.com>. It features a large "TRY A SERVER" button. Below it, there's a section titled "How to" with three numbered steps:

1. Create a repository on GitHub (`<your-username>/<your-repo>`)
2. Create a `db.json` file
3. Visit <https://my-json-server.typicode.com/<your-username>/<your-repo>> to access your server

Below the steps, a note says "No registration. Nothing to install!"

### Example

You can check the following server as an example:  
<https://my-json-server.typicode.com/typicode/demo>

### Alpha

For the moment, the project is in **alpha** which means many things may change or break:

- URLs may change
- Service may be down
- Authentication may be added later
- etc...

Finally, we apply our json code on a server.

The screenshot shows the [My JSON Server](https://my-json-server.typicode.com) dashboard. At the top, there's a message: "Fake Online REST server for teams". Below it, two main sections are shown:

- Create a JSON file on GitHub**: Shows a screenshot of a GitHub repository at `github.com/user/repo/master/db.json` containing the following JSON:

```
{  "posts": [    {      "id": 1,      "title": "Hello"    }  ],  "profile": {    "name": "typicode"  }}
```
- Get instantly a fake server**: Shows a screenshot of a browser at `my-json-server.typicode.com/user/repo/posts/1` displaying the same JSON data as the GitHub file.

And this is json code accepted in json server.

```
Secure | https://my-json-server.typicode.com/arafa83/Food/restaurants
```

Raw    Parsed

```
[{"id": 1, "name": "KFC - Zagazig", "img": "https://europe-public.foodpanda.com/dynamic/production/eg/images/vendors/e6bi_sop.jpg?v=10150903144409", "location": "1 Talat Herb, Mosque, Qesm Then AZ Zagazig, Ash Sharqia Governorate", "phone": "01281173768", "rate": 4, "foods": [{"id": 1, "name": "Chicken Meal ( Menu 25 )", "content": "Chicken piece, small plain rice, small French fries and bun ", "salary": "EGY 21.93", "image": "https://4.bp.blogspot.com/-dje6qjD8J0w/UYP%uifgr1/AAAAAAAAXc/ti_j47VgFU/s800/raising-cares-chicken-fingers-box-83.JPG", "rate": 5}, {"id": 1, "name": "Chicken Meal ( Menu 25 )", "content": "Chicken piece, small plain rice, small French fries and bun ", "salary": "EGY 21.93", "image": "https://4.bp.blogspot.com/-dje6qjD8J0w/UYP%uifgr1/AAAAAAAAXc/ti_j47VgFU/s800/raising-cares-chicken-fingers-box-83.JPG", "rate": 5}, {"id": 2, "name": "McDonald's - Zagazig", "img": "https://europe-public.foodpanda.com/dynamic/production/eg/images/vendors/e0ut_sop.png?v=20161109104888", "location": "Tolba Awida St, Shaibet an Nakareyah, Markaz El-Dakazik, Ash Sharqia Governorate ", "phone": "01281173768", "rate": 3, "foods": [{"id": 1, "name": "Big Mac", "content": "Bacon cheeseburger with lettuce, pickles, onions, and special sauce", "salary": "EGY 21.93", "image": "https://4.bp.blogspot.com/-dje6qjD8J0w/UYP%uifgr1/AAAAAAAAXc/ti_j47VgFU/s800/raising-cares-chicken-fingers-box-83.JPG", "rate": 5}, {"id": 1, "name": "Big Mac", "content": "Bacon cheeseburger with lettuce, pickles, onions, and special sauce", "salary": "EGY 21.93", "image": "https://4.bp.blogspot.com/-dje6qjD8J0w/UYP%uifgr1/AAAAAAAAXc/ti_j47VgFU/s800/raising-cares-chicken-fingers-box-83.JPG", "rate": 5}]}]
```

## 5.3 Finally Models and Extensions

### 5.3.1 Models

a swift code but used to declare variables alone.



#### Rest model

Declare all variables that I need for my application restaurants as following.

```
import Foundation

struct Rests : Decodable {

    let id: Int
    let rate: Double
    let location: String
    let name: String
    let img: String
    let phone: String
    let foods: [Food]

}
```

#### Food model

also Declare all variables that I need for my application foods as following.

```
import Foundation

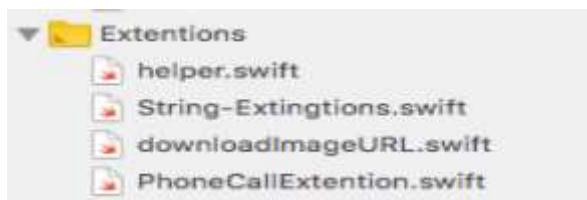
struct Food : Decodable {

    let id: Int
    let name: String
    let image: String
    let content: String
    let rate: Double
    let salary: String

}
```

### 5.3.2 Extensions

There are various extensions that provide quick access to information without necessity to open app fully, in order to make iOS app convenient for use. All you need is just to swipe the main screen left and you will see a list of Information from the database of the main application and images that were previously uploaded to the file system can be displayed here. Also, here information via the network can be uploaded. Moreover.



#### Helper extension

We use a class function to prevent us from using instance from calling.

```
import Foundation

class helper:NSObject{

    //we use class function to prevent us from using instance on calling
    class func saveToken(token : String){
        let def = UserDefaults.standard
        def.setValue(token, forKey: "api_token")
        def.synchronize()

    }
}
```

## String extinctions

Used to eliminate the spaces after and before words

```
import UIKit
//extension to eliminate the spaces after and before words
extension String{
    var trimmed:String{
        return self.trimmingCharacters(in: .whitespacesAndNewlines)
    }
}
```

## Downloading Image URL Extension

For downloading image from URL

```
import UIKit

//Extension from downloading Image From URL
extension UIImageView {
    func downloadedFrom(url: URL, contentMode mode: UIViewContentMode = .scaleAspectFit) {
        contentMode = mode
        URLSession.shared.dataTask(with: url) { [self] data, response, error in
            guard
                let httpURLResponse = response as? HTTPURLResponse, httpURLResponse.statusCode == 200,
                let mimeType = response?.mimeType, mimeType.hasPrefix("image"),
                let data = data, error == nil,
                let image = UIImage(data: data)
            else { return }
            DispatchQueue.main.async() {
                self.image = image
            }
        }.resume()
    }
    func downloadedFrom(link: String, contentMode mode: UIViewContentMode = .scaleAspectFit) {
        guard let url = URL(string: link) else { return }
        downloadedFrom(url: url, contentMode: mode)
    }
}
```

```
//load image from url
extension UIImageView {

    func setCustomImage(_ imgURLString: String?, contentMode mode: UIViewContentMode = .scaleToFill , scale : CGFloat) {
        guard let imageURLString = imgURLString else {
            self.image = UIImage(named: "default.png")
            return
        }
        DispatchQueue.global().async {
            let data = try? Data(contentsOf: URL(string: imageURLString)!)
            DispatchQueue.main.async {
                self.image = data != nil ? UIImage(data: data!) : UIImage(named: "default.png")
                self.image = self.image?.resized(withPercentage: scale)
            }
        }
    }

    /*Extension Usage

    myImageView.setCustomImage("url")*/



    extension UIImage {
        func resized(withPercentage percentage: CGFloat) -> UIImage? {
            let canvasSize = CGSize(width: size.width * percentage, height: size.height * percentage)
            UIGraphicsBeginImageContextWithOptions(canvasSize, false, scale)
            defer { UIGraphicsEndImageContext() }
            draw(in: CGRect(origin: .zero, size: canvasSize))
            return UIGraphicsGetImageFromCurrentImageContext()
        }

        func resized(toWidth width: CGFloat) -> UIImage? {
            let canvasSize = CGSize(width: width, height: CGFloat(ceil(width/size.width * size.height)))
            UIGraphicsBeginImageContextWithOptions(canvasSize, false, scale)
            defer { UIGraphicsEndImageContext() }
            draw(in: CGRect(origin: .zero, size: canvasSize))
            return UIGraphicsGetImageFromCurrentImageContext()
        }
    }
}
```

## Phone Call Extension

Applying a call to restaurants management to confirm customer food order.

```
import UIKit

extension String {

    enum RegularExpressions: String {
        case phone = "+\\(\\s*(?:\\s*\\+?((\\d{3},?)?)?(-, )*(\\d{3}))[-, ]*)*?((\\d{3})[-, ]*(\\d{2,4}))?:(-x )*(\\d+)?)\\s*#"
    }
}

func isValid(regex: RegularExpressions) -> Bool {
    return isValid(regex: regex.rawValue)
}

func isValid(regex: String) -> Bool {
    let matches = range(of: regex, options: .regularExpression)
    return matches != nil
}

func onlyDigits() -> String {
    let filteredUnicodeScalars = unicodeScalars.filter(CharacterSet.decimalDigits.contains($0))
    return String(String.UnicodeScalarView(filteredUnicodeScalars))
}

func makeACall() {
    if isValid(regex: .phone) {
        if let url = URL(string: "tel://\(self.onlyDigits())"),
           UIApplication.shared.canOpenURL(url) {
            #available(iOS 10, *)
            UIApplication.shared.open(url)
        } else {
            UIApplication.shared.openURL(url)
        }
    }
}
```

# References

All references we needed or help us in our project all of them are websites and pages.

As

- Wikipedia site.

[https://en.wikipedia.org/wiki/Main\\_Page](https://en.wikipedia.org/wiki/Main_Page)

- Msdn.microsoft site.

[https://msdn.microsoft.com/en-us/library/aa288436\(v=vs.71\).aspx](https://msdn.microsoft.com/en-us/library/aa288436(v=vs.71).aspx)

- Stackoverflow site.

<http://stackoverflow.com/questions/tagged/c%23>

- Tutorialspoint site.

<http://www.tutorialspoint.com/csharp/>

- Aforgenet Library.

<http://www.aforgenet.com/>

- GitHub

<http://www.github.com/>

- firebase

<http://www.firebaseio.com/>

