**MOBILE APPS CREATION USING PYTHON**

**Kivy – Cross-platform Python GUIs**

[**Kivy**](https://kivy.org/) an open-source Python library for developing cross-platform GUI applications. It allows you to write pure-Python graphical applications that run on the main desktop platforms (Windows, Linux, and macOS) and on iOS & Android.

Now, every time I hear about a new GUI toolkit I always want to know how “native” it feels—I believe that great GUI application should play to the strengths of the platform they run on.

Official Website: [**https://kivy.org**](https://kivy.org/)

## The BeeWare Project – Native Python Mobile Apps

The second Python GUI and mobile development framework I want to tell you about is called the [“**BeeWare**” project](https://pybee.org/project/using/). It offers you a set of tools and an abstraction layer you can use to write native-looking mobile and desktop applications using Python.

The key difference between **Kivy** and **BeeWare** is that BeeWare programs use the native UI toolkit of the platform they run on, whereas Kivy apps use a custom UI toolkit that uses the same controls across all platforms.

With BeeWare, the UI controls your app uses will be the buttons, check boxes, and form elements provided by the underlying operating system. This means you can build apps that look and feel 100% native to each specific mobile (and desktop) platform.

## Official Website: [https://beeware.org](https://beeware.org/)

## Developing Android apps completely in Python

The goal of this training is to show you how to start developing full Android applications using only Python. Different technologies will be demonstrated, including PySide-based QML GUIs using the Necessitas Qt port and the Py4A/SL4A-based approach, which can be combined with Android’s WebKit and re-use Python web frameworks.

While Android already has a good SDK out of the box, being able to use Python instead of Java is a big advantage for some developers - it allows for quicker turnaround times, and reuse of Python libraries. Python on Android uses a native CPython build, so its performance and compatibility is very good. Combined with PySide (which uses a native Qt build) and Qt’s support for OpenGL ES acceleration, you can create fluent UIs even with Python.

The resulting PySide-based applications run on Android, but also at least on Mac OS X, Windows, Linux and Maemo and MeeGo - basically all platforms on which Qt is available. The SL4A/Py4A-based applications will run on Android only, but will be able to utilize Android-specific APIs. Both approaches can also be combined.

Video link: <https://www.youtube.com/watch?v=bXrtX6yRJJI&feature=youtu.be>

## Examples of Python Based applications:

* **Aarlogic C05/3:** Ready-to-use GSM/GPS tracking PCB with a Python development board along with the support of test server based on Google Maps.
* **Pyroute:**A GPS-capable mapping/routing application for mobile.
* **FoodPlus:**A mobile food app which simplifies the process of food ordering and tracking, specially designed for food lovers.
* **AppBackup:** An app for jailbroken iOS devices that lets one back up and restore the settings and data of App Store apps.

**Developing Mobile app using React Js:**

**React** is enabling frontend developers to build apps like never before. It's benefits are many: one-way data flow, easy component lifecycle methods, declarative components and more. [**Reapp**](http://reapp.io/) was recently released on React. It's a mobile app platform designed for performance and productivity. Think of it as a well-optimized UI kit, along with a build system and a bunch of helpers that let you build apps easily.

[React.js : A Framework That Can Give Your App Wings](https://www.consagous.com/develop-your-own-mobile-applications-with-react-js/)

We are intrigued by frameworks that introduce new ways of thinking in web development by opening the door of new opportunities. That’s the reason, we include React.js framework is our app development process. It is a JavaScript library which was introduced by Facebook on 4th March, 2013.Today, React.js has become one of the most fiscal framework of JavaScript and most of the popular websites including Imgur,SeatGeek, Netflix, Airbnb,HelloSign and others use ReactJS.  
 

Benefits of using React.js

We know that modifying and querying Document Object Model could be a major performance bottleneck in web mobile application development. If I had to single out one reason to use React it might be precisely the optimizations in that namespace.  
   
Although React is quick with DOM operations and comes with no bundled UI components. This makes it on par with Angular and Backbone, each of that I have already given with an adequate UI bundle. During this review I won’t target the UI as most of the UI – Only frameworks depend on third-party JavaScript.

### MV\* Pattern

Here comes a shocker – however, keep on reading – React isn’t AN MVC framework! Optimized for one-way binding (or unidirectional data flow) Models in the sense notable to Backbone or Sencha touch aren’t present in React. Fear not, two-way binding primarily happens in sort of events triggered by an amendment in part state. Here we introduced an element, that may be a mixture of view and Controller. With this notion in mind, it’s hard not to notice that something almost revolutionary is going on here. I will be able to discuss elements any down within the DOM section.

### Class System

ReactJS is comes up with a class system as everything revolves around the elements. In this framework, elements have their own ecosystem which are mainly developed with a core concept but this is an unexceptional method of React.createClass. Since the conventional MVC (Model-View Controller) is absent in React.js therefore its traditional class system become too outdated.

### DOM Control

React is a library for building User Interfaces, however, approaches things differently by breaking them into components. Rather than classic templates, React framework offers JSX, AN XML based templates that compiles to JavaScript.Once compiled into JavaScript, it creates a light-weight representation of the document.

### Third party plugins

React.js does not provide third party plugins support therefore it is quite difficult to combine it with Angular, Backbone and even ClojureScript.However, you have to keep yourself update about the latest releases of plugins.

### Documentation

If you are using React to develop the mobile application, then you should familiar with documentation as it will help you to finish your task as soon as possible. React.js documentation contain guides and tips that will help you to start building app and also provide some external resources for future reference.

## Recommended Operating Systems

* **Windows:**7 or newer
* **MAC:** OS X v10.7 or higher
* **Linux:** Ubuntu

## Hardware Requirements

We strongly recommend a computer fewer than 5 years old.

* Processor: Minimum 1 GHz; Recommended 2GHz or more
* Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
* Hard Drive: Minimum 32 GB; Recommended 64 GB or more
* Memory (RAM): Minimum 1 GB; Recommended 4 GB or above
* Sound card w/speakers
* Some classes require a camera and microphone

## Recommended Software

### Supported Browsers

People often ask what browser they should use. There is no single answer for this. Use whichever browser works best on your computer. However, we recommend downloading Firefox and/or Chrome in addition to having Internet Explorer or Safari.

* [Firefox](http://www.mozilla.com/)
* [Chrome](http://www.google.com/chrome/)

**Questionarie:**

## What are the features or functionalities you’re looking to integrate into your mobile app? Please explain as detailed as possible?

## Would you like to have a CMS for the app? (CMS is a system which lets you add content and make day-to-day updates on your mobile app) [This is important as we’ll have to leave a framework considering the future additions while coding the app?

## Will the solution require in-app purchase, payment gateway integration, multi-currency support, etc?

## How would you want the app to be developed? Do you prefer using Cross Platform technologies or developing the apps natively across each platform?

## What are the desired compatible devices & OS versions? Please provide required platform and OS versions. (Eg: iOS, Android, Blackberry, Windows) [It is recommended to develop apps in the latest OS versions.]?

## Is the app expected to work in both Portrait and Landscape orientation?

## What languages does the app need to support?

## Would you like to have the app listed as free or paid?

## Can the app be used offline (without internet connection)?

## If you’re providing us with web services, please indicate the API format which it’ll be provided. An API guide would be highly appreciated.?

## Would you like to integrate or use any third party API services for your application? ( For Eg: Foursquare, Social network authentication such as Facebook, Twitter, etc.)?