# Applications Development and Emerging Technologies [1]

## A Study Guide for Students of Sorsogon State University - Bulan Campus[2]

Jarrian Vince G. Gojar[3]

January 23, 2025

[1] A course in the Bachelor of Science in Computer Science.

[2] This book is a study guide for students of Sorsogon State University - Bulan Campus taking up the course Applications Development and Emerging Technologies.

[3] https://github.com/godkingjay

Sorsogon State University - Bulan Campus

# Contents

# List of Figures

# List of Tables

# List of Codes

# Preface

–

Jarrian Vince G. Gojar
https://github.com/godkingjay

# 1

# Introduction to Applications Development and Emerging Technologies

## 1.1 Introduction

### 1.1.1 Application

An **application** is a software program that allows users to perform specific tasks. Applications for desktop or laptop computers are sometimes called **desktop applications**, while those for mobile devices are called **mobile apps**. When you open an application, it runs inside the operating system until you close it. Most of the time, you will have more than one application open at the same time, which is known as **multitasking**.

### 1.1.2 Development

**Development** is the process of creating a software application. It includes designing the user interface, writing code, and testing the application for bugs. The goal of software development is to create a program that is easy to use and works correctly.

### 1.1.3 Application Development

**Application development** is the process of planning, designing, creating, testing, and deploying an application to perform various business operations. It can be done by massive organizations with large teams working on projects or by a single freelance developer.

## 1.2 Different Types of Application Development

There are several different types of application development, including:

- Web Development
- Mobile Application Development
- Desktop Application Development
- Game Development
- Cloud Development

### 1.2.1 Web Development

**Web development** is the process of creating websites and web applications. It involves designing the user interface, writing code, and testing the website for bugs. Web development can be divided into two categories: front-end development and back-end development.

#### 1.2.1.1 Front-End Development

**Front-end development** is the process of creating the user interface of a website. It involves designing the layout, colors, and fonts of the website. Front-end developers use HTML, CSS, and JavaScript to create the user interface of a website.

1. **HTML (HyperText Markup Language)** is the standard markup language used to create web pages. It defines the structure of a web page using a series of elements.

2. **CSS (Cascading Style Sheets)** is a style sheet language used to define the appearance of a web page. It allows developers to control the layout, colors, and fonts of a website.

3. **JavaScript** is a programming language used to create interactive elements on a web page. It allows developers to add functionality such as animations, pop-ups, and form validation to a website.

4. **Bootstrap** is a front-end framework that allows developers to create responsive and mobile-first websites. It provides a set of pre-designed components, such as buttons, forms, and navigation bars, that can be easily customized.

5. **React** is a JavaScript library used to create user interfaces for single-page applications. It allows developers to build reusable components that update automatically when the data changes.

6. **Angular** is a front-end framework that allows developers to create dynamic web applications. It provides a set of tools and libraries for building interactive user interfaces.

7. **Vue** is a progressive JavaScript framework used to create user interfaces and single-page applications. It allows developers to build interactive web applications with ease.

The above tools and technologies are commonly used in front-end development to create responsive and interactive websites. HTML, CSS, and JavaScript are the building blocks of front-end development, while frameworks such as Bootstrap, React, Angular, and Vue provide additional features for building modern web applications.

#### 1.2.1.2 Back-End Development

**Back-end development** is the process of creating the server-side logic of a website. It involves writing code that interacts with the database and processes data. Back-end developers use programming languages such as PHP, Python, and Ruby to create the server-side logic of a website.

1. **Node.js** is a JavaScript runtime environment that allows developers to run JavaScript on the server-side. It provides a set of libraries and tools for building scalable and high-performance web applications.

2. **Express** is a web application framework for Node.js. It provides a set of features for building web applications, such as routing, middleware, and templating.

3. **Django** is a high-level web framework for Python. It allows developers to build web applications quickly and efficiently. Django provides a set of tools and libraries for building secure and scalable web applications.

4. **Flask** is a lightweight web framework for Python. It allows developers to build web applications with minimal code. Flask provides a set of tools and libraries for building simple and scalable web applications.

5. **Ruby on Rails** is a web application framework for Ruby. It provides a set of tools and libraries for building web applications quickly and efficiently. Ruby on Rails follows the convention over configuration principle, which allows developers to write less code and focus on building the application.

6. **Laravel** is a web application framework for PHP. It provides a set of tools and libraries for building web applications quickly and efficiently. Laravel follows the model-view-controller (MVC) architecture, which allows developers to separate the business logic from the presentation layer.

7. **Spring** is a web application framework for Java. It provides a set of tools and libraries for building enterprise-level web applications. Spring follows the inversion of control (IoC) principle, which allows developers to write loosely coupled code and focus on building the application.

The above tools and technologies are commonly used in back-end development to create the server-side logic of a website. Back-end developers use these tools to interact with the database, process data, and handle user requests on the server-side.

### 1.2.2 Mobile Application Development

**Mobile application development** is the process of creating mobile applications for smartphones and tablets. It involves designing the user interface, writing code, and testing the mobile application for bugs. Mobile development can be divided into two categories: iOS development and Android development.

#### 1.2.2.1 iOS Development

**iOS development** is the process of creating mobile applications for Apple devices, such as iPhones and iPads. It involves designing the user interface using Xcode and writing code in Swift or Objective-C. iOS developers use Xcode, Swift, and Objective-C to create mobile applications for Apple devices.

1. **Xcode** is an integrated development environment (IDE) used to create iOS applications. It provides a set of tools and libraries for building mobile applications for Apple devices.

2. **Swift** is a programming language used to create iOS applications. It provides a set of features for building mobile applications, such as type safety, optionals, and generics.

3. **Objective-C** is a programming language used to create iOS applications. It provides a set of features for building mobile applications, such as dynamic typing, message passing, and memory management.

4. **React Native** is a JavaScript framework used to create mobile applications for Android and iOS devices. It allows developers to build cross-platform mobile applications with a single codebase.

5. **Flutter** is a mobile UI framework used to create mobile applications for Android and iOS devices. It allows developers to build cross-platform mobile applications with a single codebase.

The above tools and technologies are commonly used in iOS development to create mobile applications for Apple devices. iOS developers use these tools to design the user interface and write code for mobile applications.

#### 1.2.2.2 Android Development

**Android development** is the process of creating mobile applications for Android devices. It involves designing the user interface using Android Studio and writing code in Java or Kotlin. Android developers use Android Studio, Java, and Kotlin to create mobile applications for Android devices.

1. **Android Studio** is an integrated development environment (IDE) used to create Android applications. It provides a set of tools and libraries for building mobile applications for Android devices.

2. **Java** is a programming language used to create Android applications. It provides a set of features for building mobile applications, such as object-oriented programming, inheritance, and polymorphism.

3. **Kotlin** is a programming language used to create Android applications. It provides a set of features for building mobile applications, such as null safety, extension functions, and coroutines.

4. **React Native** is a JavaScript framework used to create mobile applications for Android and iOS devices. It allows developers to build cross-platform mobile applications with a single codebase.

5. **Flutter** is a mobile UI framework used to create mobile applications for Android and iOS devices. It allows developers to build cross-platform mobile applications with a single codebase.

The above tools and technologies are commonly used in Android development to create mobile applications for Android devices. Some of the tools here are also used in iOS development to create mobile applications for Apple devices. React Native and Flutter in particular are used to build cross-platform mobile applications for both Android and iOS devices.

### 1.2.3 Desktop Application Development

**Desktop application development** is the process of creating desktop applications for Windows, macOS, and Linux. It involves designing the user interface, writing code, and testing the desktop application for bugs.

1. **Electron** is a framework used to create desktop applications with web technologies. It allows developers to build cross-platform desktop applications with HTML, CSS, and JavaScript.

2. **JavaFX** is a framework used to create desktop applications with Java. It provides a set of tools and libraries for building cross-platform desktop applications with Java.

3. **Qt** is a framework used to create desktop applications with C++. It provides a set of tools and libraries for building cross-platform desktop applications with C++.

4. **WinForms** is a framework used to create desktop applications with C#. It provides a set of tools and libraries for building desktop applications for Windows.

5. **WPF** is a framework used to create desktop applications with C#. It provides a set of tools and libraries for building desktop applications for Windows.

The above tools and technologies are commonly used in desktop development to create desktop applications for Windows, macOS, and Linux. For Windows, developers use WinForms and WPF to create desktop applications with C#. For cross-platform desktop applications, developers use Electron, JavaFX, and Qt to build desktop applications with web technologies, Java, and C++.

### 1.2.4 Game Development

**Game development** is the process of creating video games for consoles, computers, and mobile devices. It involves designing the gameplay, writing code, and testing the game for bugs.

1. **Unity** is a game engine used to create 2D and 3D games for consoles, computers, and mobile devices. It provides a set of tools and libraries for building cross-platform games with C#.

2. **Unreal Engine** is a game engine used to create 2D and 3D games for consoles, computers, and mobile devices. It provides a set of tools and libraries for building cross-platform games with C++.

3. **Godot** is a game engine used to create 2D and 3D games for consoles, computers, and mobile devices. It provides a set of tools and libraries for building cross-platform games with GDScript.

4. **GameMaker Studio** is a game engine used to create 2D games for consoles, computers, and mobile devices. It provides a set of tools and libraries for building cross-platform games with GML.

5. **Construct** is a game engine used to create 2D games for consoles, computers, and mobile devices. It provides a set of tools and libraries for building cross-platform games with events.

The above tools and technologies are commonly used in game development to create video games for consoles, computers, and mobile devices. Unity, Unreal Engine, Godot, GameMaker Studio, and Construct are popular game engines used by game developers to create 2D and 3D games. These game engines provide a set of tools and libraries for building cross-platform games with C#, C++, GDScript, and GML.

### 1.2.5 Cloud Development

**Cloud development** is the process of creating cloud-based applications that run on remote servers. It involves designing the user interface, writing code, and testing the cloud application for bugs.

1. **Amazon Web Services (AWS)** is a cloud platform used to create cloud-based applications. It provides a set of tools and services for building scalable and secure cloud applications.

2. **Microsoft Azure** is a cloud platform used to create cloud-based applications. It provides a set of tools and services for building scalable and secure cloud applications.

3. **Google Cloud Platform (GCP)** is a cloud platform used to create cloud-based applications. It provides a set of tools and services for building scalable and secure cloud applications.

4. **Heroku** is a cloud platform used to create cloud-based applications. It provides a set of tools and services for building scalable and secure cloud applications.

5. **Firebase** is a cloud platform used to create cloud-based applications. It provides a set of tools and services for building scalable and secure cloud applications.

The above tools and technologies are commonly used in cloud development to create cloud-based applications that run on remote servers. AWS, Microsoft Azure, GCP, Heroku, and Firebase are popular cloud platforms used by developers to build scalable and secure cloud applications. These cloud platforms provide a set of tools and services for building cloud-based applications with ease.

# 2

# Web Development

## 2.1 Introduction

There are around 3.58 billion internet users on the planet. This implies that over half of the world's 7.6 billion people have access to the internet, which they use for everything from entertainment to education, communication to commerce, keeping up with current events, and keeping up with business experts. Indeed, for most people, the internet is the first (and often only) channel through which we communicate with the world in all of its complexities.

There are three interactive elements on the internet:

1. **Websites** - A collection of web pages that are linked together and share a common domain name.
2. **Servers** - A computer or computer program that manages access to a centralized resource or service in a network.
3. **Browsers** - A software application used to access and view websites on the internet.

The frontend (client side) and the backend (server side) are two parts of any website. The frontend comprises everything the user sees and experiences instantly while visiting a website. The backend is behind the scenes that store, send and receive information.

HTML, CSS, and Javascript files make up everything a user sees on a website. As a web developer, these are the most basic tools needed. They are the languages that required to build websites.

## 2.2 HTML

**HTML (HyperText Markup Language)** is the standard markup language used to create web pages. It defines the structure of a web page using a series of elements. It contains the essential elements of a website, such as words, titles, and p aragraphs, as well as links, images, and other media. HTML elements are represented by tags, which are enclosed in angle brackets.

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
```

```
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>First Web Page</title>
</head>

<body>
    <h1>Hello, World!</h1>
    <p>Welcome to my website.</p>
</body>

</html>
```

Code 2.1: HTML Example

Code 2.1 shows an example of an HTML document. The `<!DOCTYPE html>` declaration defines the document type and version of HTML. The `<html>` element contains the entire HTML document. The `<head>` element contains metadata about the document, such as the title of the page. The `<body>` element contains the content of the page, such as headings, paragraphs, and other elements.

# 3

# Version Control

# 4

# NextJS

# 5

# Mobile Applications Development

# 6

# Cloud Computing

# 7

# Artificial Intelligence

# 8

# Internet of Things and Augmented Reality

**A. Books**

- 

**B. Other Sources**

- 