**Project Title: RoomNinja**

**1. INTRODUCTION:**

Project is constituent of seven words and I have tried my best to give its definition analyzing each word in the following manner:

P- Perfect Planning

R- Resources

O- Organization

J- Joint Efforts

E- Engineering Skill

C- Communication

T- Technique

**RoomNinja:**

RoomNinja is a PG search web app online platform, specifically designed to help users search and find Paying Guest accommodations or shared living spaces. It offers a user-friendly interface accessible through a web browser, allowing individuals to conveniently search and explore available PG options.

The web app typically provides various features and functionalities to enhance the user experience. These may include advanced search filters to narrow down preferences such as location, budget, amenities, and room types. The app may also display detailed property listings, including information about facilities, nearby amenities, and photos of the accommodations.

Users can typically view contact details of landlords or property managers, allowing them to directly connect and inquire about the availability and terms of the PG accommodations. Some web apps may also offer additional features like online booking, virtual tours, reviews and ratings, and interactive maps to assist users in making informed decisions.

The primary advantage of a PG search web app is the accessibility it offers across different devices with an internet connection. Users can access the app from their desktops, laptops, tablets, or smartphones, providing flexibility and convenience in their search for suitable PG accommodations.

Overall, a PG search web app simplifies and streamlines the process of finding PG accommodations by providing a centralized platform with comprehensive search capabilities, enabling users to efficiently explore and connect with potential housing options.The main objective of the project on E-Document:

* User-friendly interface accessible via web browsers.
* Advanced search filters to refine preferences (location, budget, amenities, room types).
* Detailed property listings with comprehensive information and photos.
* Contact details of landlords or property managers for direct communication.
* Online booking functionality for convenient reservation.
* Virtual tours or interactive maps to visualize the accommodations.
* Reviews and ratings to gauge the quality and experiences of past occupants.
* Availability updates to ensure real-time information.
* Accessibility across devices (desktops, laptops, tablets, smartphones).
* Streamlined and centralized platform for efficient PG accommodation search.
* Time-saving by providing a wide range of options in one place.
* Facilitates informed decision-making through detailed information.
* Convenience in comparing different PG accommodations.
* Flexibility to explore and connect with potential housing options.

**2. PROJECT DESCRIPTION:**

**Modules of Project**

There are various modules associate with project. These modules are working in their specific area to lead and complete the project.

**Admin Modules of RoomNinja:**

* Login:

The login is used to sign in the application. The login activity contains username and password text field and login button for login process.

* Manage PG/Rooms:

In this part of module, admin add PG’s record into the database. The PG’s record includes rooms, location, category etc.

* Manage Booking:

In this part of module, admin add booking’s record into the database. The Booking’s record includes PG-Name, id, city, . By adding teacher, the admin gives permission to teacher to access the site.

* ManageDashboard**:**

This is almost similar as the above one but the difference is here we are dealing with view what user will see or what user can’t.

* Manage Feedback:

The admin can add, update and delete various types of Feedback. All the student can have view announcements.

* Manage Contact:

Here admin can view and response to the contact.

**User Module:**

* Register**:** The very first step is Register in the app. After the successful registration user can login in the app.
* Login:

The Second step is login. Firstly, the use will login into the site with his/her email-id and password .

* Dashboard:

All the user can have view their desired PG online and can search the PG availability according to their desired location. They can make of request for PG using dashboard section.

* Feedback:

User can write feedback so admin can make changes according to the requirements provided.

* Booking:

User can view their bookings and can make requests for new PG.

**3. ANALYSIS:**

**Problem Description:**

**3.1 Existing System:**

1. Classified Websites: Many online classified websites, such as Craigslist, Sulekha, and OLX, have sections dedicated to rental accommodations, including paying guest options. Users can search for PG listings based on location, budget, and other preferences.
2. Real Estate Portals: Some real estate portals, like and 99acres, also feature PG listings alongside other rental properties. These portals often provide filters to narrow down search results based on factors like location, price range, and amenities.
3. Local Advertisements: Traditional methods, such as newspaper classifieds or community notice boards, continue to be used for advertising PG accommodations. People can find relevant advertisements in local newspapers or on community bulletin boards.
4. Word of Mouth: Friends, colleagues, or acquaintances who have knowledge of available PG accommodations may refer or recommend suitable options to others. This informal network can be a valuable source of information for finding PG accommodations.

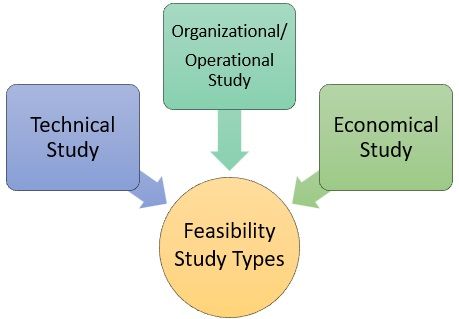
**3.2 Proposed System:**

1. Developing a mobile app dedicated to PG searches would make it more convenient for users to access and search for PG accommodations. The app could offer features like GPS-based location search, user reviews and ratings, and instant notifications for new listings.
2. The proposed system could utilize machine learning algorithms to analyze user preferences and provide personalized recommendations for suitable PG accommodations based on factors like location, budget, amenities, and user reviews.

**3.3 Feasibility Study:**

A feasibility study is a preliminary study which investigates the information needs of perspective users and determines the resource requirements, determining the cost effectiveness of various alternatives in the designs of the information system, benefits and feasibility of proposed project.

The goal of the feasibility study is to evaluate alternative systems to propose the most feasible and desirable systems for development.



**Types of Feasibility:**

There are various measures of feasibility that helps to decide whether a particular project is feasible or not. These measures include-

**Technical Feasibility Study -**

The technical issues raised during the technical feasibility analysis are:

* Does the necessary technology exist to do what is suggested?
* Does the proposal equipment have the technical capacity to hold the data required to use the new system?
* Will the proposed system & components provide adequate responses to inquiries, regardless of the number or locations of users?
* Can the system be expanded?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using Java the project is technically feasible for development.

**Operational Feasibility Study -**

We have designed front end in XML by getting the information from the end user, which help us in designing the GUI according to the end user’s requirements. The end users can easily understand and expand it in the future.

This includes the following questions:

* Is there sufficient support for the users?
* Will the proposed system cause harm?
* The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

**Economic Feasibility Study -**

It involves estimating cost and benefits that can be tangible and intangible because of confusing between the types of costs it is sometimes very difficult to divide the benefits out weight the cost.

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

The costs conduct a full system investigation.

* The cost of the hardware and software.
* The costs conduct a full system investigation.
* The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

**3.4 Requirement Analysis:**

**SYSTEM REQUIRENMENT SPECIFICATIONS**

A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform.

An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety of real-world situations. Parameters such as operating speed, response time, availability, portability, maintainability, footprint, security and speed of recovery from adverse events are evaluated. Methods of defining an SRS are described by the IEEE (Institute of Electrical and Electronics Engineers) specification 830-1998.



**Need for SRS:**

A basic purpose of software requirements specification is to bridge the communication gap between the user & system analyst, SRS is the medium through which the client and user needs are accurately specified; indeed SRS forms the basis of software development.  A good SRS should satisfy all the parties- sometimes very hard to achieve and involves trade off and persuasion.  Another important purpose of developing an SRS is helping the clients to understand their own needs.

A main purpose of the product specification is to define the need of the product’s user. Sometimes, the specification may be a part of a contract sign between the producer and the user. It could also form part of the user manuals. A   user‘s needs are sometimes not clearly understood by the developer. If this is the case, a careful analysis – involving much interaction with the user should be devoted to reaching a clear statement of requirements, in order to avoid possible misunderstandings. Sometimes, at the beginning of a project, even the user has no clear idea of what exactly the desired product is. Think for instance of user interface , a user with no previous experience with computer products may not appreciate the difference  between , say menu driven interaction and a command line interface. Even an exact formation of system functions and performance may be missing an initial description produced by an experienced user.

**4. Hardware and Software Requirements**

For this project minimum hardware and software requirement are listed below:

Minimum Hardware Requirements:

Processor: i3

RAM: 8 GB

SSD: 256GB

Software Requirements:

Front End: HTML, CSS, Bootstrap, JavaScript, ECMA Script, React JS

DB Tool: Firestore

Browser: Mozilla Firefox/Chrome/Edge

OS: Windows Operating System/Linux

Text Editor: Visual Studio Code

Environment: Firebase

**Software Tool Theoretical Background**

**Firebase:**

-Authentication: Utilize Firebase Authentication to handle user authentication and secure access to backend services.

- Real-time Database: Utilize Firebase Realtime Database to store and retrieve data in real-time, such as user profiles, PG listings, and booking details.

- Cloud Firestore: Use Firebase Cloud Firestore, a NoSQL document database, for structured data storage and querying capabilities for features like user reviews and ratings.

- Cloud Functions: Implement Cloud Functions for Firebase to run server-side code and perform backend logic, such as sending notifications, handling booking requests, and updating data.

- Storage: Utilize Firebase Storage for storing and serving user-uploaded images or other media associated with PG listings.

- Push Notifications: Utilize Firebase Cloud Messaging (FCM) to send push notifications to users, such as updates on booking status or new PG listings matching user preferences.

- Security Rules: Configure Firebase Security Rules to define granular access control and protect sensitive data, ensuring that only authorized users can perform specific actions.

- Analytics: Utilize Firebase Analytics to gain insights into user behavior, monitor usage patterns, and make data-driven decisions to improve the platform.

- Crash Reporting: Implement Firebase Crashlytics to track and report application crashes, helping identify and fix bugs promptly.

- Performance Monitoring: Utilize Firebase Performance Monitoring to monitor backend performance metrics, identify bottlenecks, and optimize API response times.

- Integration with other Firebase services: Leverage additional Firebase services like Firebase Cloud Storage, Firebase Remote Config, or Firebase Test Lab for additional functionalities or testing needs as required by the backend system.

**VS Code:**

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas.



VS Code comes with a simple and intuitive layout that maximizes the space provided for the editor while leaving ample room to browse and access the full context of your folder or project. The UI is divided into five main areas:

* **Editor** - The main area to edit your files. You can open as many editors as you like side by side vertically and horizontally.
* **Primary Side Bar** - Contains different views like the Explorer to assist you while working on your project.
* **Status Bar** - Information about the opened project and the files you edit.
* **Activity Bar** - Located on the far left-hand side, this lets you switch between views and gives you additional context-specific indicators, like the number of outgoing changes when Git is enabled.
* **Panel** - An additional space for views below the editor region. By default, it houses output, debug information, errors and warnings, and an integrated terminal. Panel can also bemoved to the left or right for more vertical space.

## [Keyboard shortcuts](https://code.visualstudio.com/docs/editor/codebasics#_keyboard-shortcuts)



Being able to keep your hands on the keyboard when writing code iscrucial for high productivity. VS Code has a rich set of default keyboard shortcuts as well as allowing you to customize them.

* [Keyboard Shortcuts Reference](https://code.visualstudio.com/docs/getstarted/keybindings#_keyboard-shortcuts-reference) - Learn the most commonly used and popular keyboard shortcuts by downloading the reference sheet.
* [Install a Keymap extension](https://code.visualstudio.com/docs/getstarted/keybindings#_keymap-extensions) - Use the keyboard shortcuts of your old editor (such as Sublime Text, Atom, and Vim) in VS Code by installing a Keymap extension.
* [Customize Keyboard Shortcuts](https://code.visualstudio.com/docs/getstarted/keybindings#_keyboard-shortcuts-editor) - Change the default keyboard shortcuts to fit your style

**HTML**

### What is HTML?

Okay, so this is the only bit of mandatory theory. In order to begin to write HTML, it helps if you know what you are writing.

HTML is the **language in which most websites are written**. HTML is used to create pages and make them functional.

The code used to make them visually appealing is known as CSS and we shall focus on this in a later tutorial. For now, we will focus on **teaching you how to build rather than design**.

### The History of HTML

HTML was first created by Tim Berners-Lee, Robert Cailliau, and others starting in **1989**. It stands for Hyper Text Markup Language.

Hypertext means that the document contains **links that allow the reader to jump to other places** in the document or to another document altogether. The latest version is known as [HTML5](https://html.com/html5/).

A **Markup Language** is a way that computers speak to each other to control how text is processed and presented. To do this HTML uses two things: tags and **attributes**.

### What are Tags and Attributes?

Tags and attributes are the basis of HTML.

They work together but perform different functions – it is worth investing 2 minutes in **differentiating the two**.

#### What Are HTML Tags?

[Tags](https://html.com/tags/) are used to **mark up the start of an HTML element** and they are usually enclosed in angle brackets. An example of a tag is: <h1>.

Most tags must be opened <h1> and closed </h1> in order to function.

#### What are HTML Attributes?

[Attributes](https://html.com/attributes/) contain **additional pieces of information**. Attributes take the form of an opening tag and additional info is **placed inside**.

An example of an attribute is:

<img src="mydog.jpg" alt="A photo of my dog.">

| **Element** | **Meaning** | **Purpose** |
| --- | --- | --- |
| **<b>** | Bold | Highlight important information |
| **<strong>** | Strong | Similarly to bold, to highlight key text |
| **<i>** | Italic | To denote text |
| **<em>** | Emphasised Text | Usually used as image captions |
| **<mark>** | Marked Text | Highlight the background of the text |
| **<small>** | Small Text | To shrink the text |
| **<strike>** | Striked Out Text | To place a horizontal line across the text |
| **<u>** | Underlined Text | Used for links or text highlights |
| **<ins>** | Inserted Text | Displayed with an underline to show an inserted text |
| **<sub>** | Subscript Text | Typographical stylistic choice |
| **<sup>** | Superscript Text | Another typographical presentation style |

**CSS**

As we have mentioned before, CSS is a language for specifying how documents are presented to users — how they are styled, laid out, etc.

A document is usually a text file structured using a markup language — HTML is the most common markup language, but you may also come across other markup languages such as SVG or XML.

Presenting a document to a user means converting it into a form usable by your audience. Browsers, like Firefox, Chrome, or Edge, are designed to present documents visually, for example, on a computer screen, projector, or printer.

## [Adding CSS to our document](https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/Getting_started#adding_css_to_our_document)

The very first thing we need to do is to tell the HTML document that we have some CSS rules we want it to use. There are three different ways to apply CSS to an HTML document that you'll commonly come across, however, for now, we will look at the most usual and useful way of doing so — linking CSS from the head of your document.

Create a file in the same folder as your HTML document and save it as styles.css. The .css extension shows that this is a CSS file.

To link styles.css to index.html, add the following line somewhere inside the [<head>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/head) of the HTML document:

HTMLCopy to Clipboard

<link rel="stylesheet" href="styles.css" />

This [<link>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/link) element tells the browser that we have a stylesheet, using the rel attribute, and the location of that stylesheet as the value of the href attribute. You can test that the CSS works by adding a rule to styles.css. Using your code editor, add the following to your CSS file:

CSSCopy to Clipboard

h1 {

color: red;

}

Save your HTML and CSS files and reload the page in a web browser. The level one heading at the top of the document should now be red. If that happens, congratulations — you have successfully applied some CSS to an HTML document. If that doesn't happen, carefully check that you've typed everything correctly.

You can continue to work in styles.css locally, or you can use our interactive editor below to continue with this tutorial. The interactive editor acts as if the CSS in the first panel is linked to the HTML document, just as we have with our document above.

## [Styling HTML elements](https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/Getting_started#styling_html_elements)

By making our heading red, we have already demonstrated that we can target and style an HTML element. We do this by targeting an element selector — this is a selector that directly matches an HTML element name. To target all paragraphs in the document, you would use the selector p. To turn all paragraphs green, you would use:

CSSCopy to Clipboard

p {

color: green;

}

What is a selector?

A CSS selector is the first part of a CSS Rule. It is a pattern of elements and other terms that tell the browser which HTML elements should be selected to have the CSS property values inside the rule applied to them. The element or elements which are selected by the selector are referred to as the subject of the selector.

Some code with the h1 highlighted.

In other articles you may have met some different selectors, and learned that there are selectors that target the document in different ways — for example by selecting an element such as h1, or a class such as .special.

}

Type, class, and ID selectors

This group includes selectors that target an HTML element such as an <h1>.

h1 {

}

It also includes selectors which target a class:

CSS

Copy to Clipboard

.box {

}

or, an ID:

CSS

Copy to Clipboard

#unique {

}

**CSS Comments:-**

Comments are used to explain the code, and may help when you edit the source code at a later date.

Comments are ignored by browsers.

A CSS comment starts with /\* and ends with \*/. Comments can also span multiple lines:

Example

p {  
    color: red;  
    /\* This is a single-line comment \*/  
    text-align: center;  
}

**CSS Colors**

Colors in CSS are most often specified by:

* a valid color name - like "red"
* an RGB value - like "rgb(255, 0, 0)"
* a HEX value - like "#ff0000"

## RGB (Red, Green, Blue)

RGB color values can be specified using this formula: rgb(red, green, blue).

Each parameter (red, green, blue) defines the intensity of the color between 0 and 255.

## Hexadecimal Colors

RGB values c+an also be specified using **hexadecimal** color values in the form: #RRGGBB, where RR (red), GG (green) and BB (blue) are hexadecimal values between 00 and FF (same as decimal 0-255).

# CSS Backgrounds

# The CSS background properties are used to define the background effects for elements.

# CSS background properties:

* Background-color:
* Background-image:
* Background-repeat:
* Background-attachment:
* Background-position:

[**Pseudo-classes and pseudo-elements**](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Selectors#pseudo-classes_and_pseudo-elements)

* This group of selectors includes pseudo-classes, which style certain states of an element. The :hover pseudo-class for example selects an element only when it is being hovered over by the mouse pointer:
* CSSCopy to Clipboard
* a:hover {
* }
* It also includes pseudo-elements, which select a certain part of an element rather than the element itself. For example, ::first-line always selects the first line of text inside an element (a <p> in the below case), acting as if a <span> was wrapped around the first formatted line and then selected.
* CSSCopy to Clipboard
* p::first-line {
* }

### [Combinators](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Selectors#combinators)

* The final group of selectors combine other selectors in order to target elements within our documents. The following, for example, selects paragraphs that are direct children of <article> elements using the child combinator (>):
* CSSCopy to Clipboard
* article > p {
* }

**JavaScript**

JavaScript is a programming language initially designed to interact with elements of web pages. In web browsers, JavaScript consists of three main parts:

ECMAScript provides the core functionality.

The Document Object Model (DOM) provides interfaces for interacting with elements on web pages

The Browser Object Model (BOM) provides the browser API for interacting with the web browser.

JavaScript allows you to add interactivity to a web page. Typically, you use JavaScript with HTML and CSS to enhance a web page’s functionality, such as validating forms, creating interactive maps, and displaying animated charts.

When a web page is loaded, i.e., after HTML and CSS have been downloaded, the JavaScript engine in the web browser executes the JavaScript code. The JavaScript code then modifies the HTML and CSS to update the user interface dynamically.

The JavaScript engine is a program that executes JavaScript code. In the beginning, JavaScript engines were implemented as interpreters.

However, modern JavaScript engines are typically implemented as just-in-time compilers that compile JavaScript code to bytecode for improved performance.

Client-side vs. Server-side JavaScript

When JavaScript is used on a web page, it is executed in web browsers. In this case, JavaScript works as a client-side language.

JavaScript can run on both web browsers and servers. A popular JavaScript server-side environment is Node.js. Unlike client-side JavaScript, server-side JavaScript executes on the server that allows you to access databases, file systems, etc.

JavaScript History

In 1995, JavaScript was created by a Netscape developer named Brendan Eich. First, its name was Mocha. And then, its name was changed to LiveScript.

Netscape decided to change LiveScript to JavaScript to leverage Java’s fame, which was popular. The decision was made just before Netscape released its web browser product Netscape Navigator 2. As a result, JavaScript entered version 1.0.

Netscape released JavaScript 1.1 in Netscape Navigator 3. In the meantime, Microsoft introduced a web browser product called the Internet Explorer 3 (IE 3), which competed with Netscape. However, IE came with its own JavaScript implementation called JScript. Microsoft used the name JScript to avoid possible license issues with Netscape.

Hence, two different JavaScript versions were in the market:

JavaScript in Netscape Navigator

JScript in Internet Explorer

Whitespace refers to characters that provide the space between other characters. JavaScript has the following whitespace:

Carriage return

Space

New Line

tab

JavaScript engine ignores whitespace. However, you can use whitespace to format the code to make it easy to read and maintain.

The following JavaScript code doesn’t use whitespace:

let formatted = true; if (formatted) {console.log('The code is easy to read');}

Code language: JavaScript (javascript)

It’s is equivalent to the following code that uses whitespace. Hence, this code is much easy to read:

let formatted = true;

if (formatted) {

console.log('The code is easy to read');}

Code language: JavaScript (javascript)

Note that JavaScript bundlers remove all whitespace from JavaScript files and put them into a single file for deployment. By doing this, JavaScript bundlers make the JavaScript code lighter and faster to load in the web browsers.

**Statements**

A statement is a code that declares a variable or instructs the JavaScript engine to do a task. A simple statement is terminated by a semicolon (;).

Although the semicolon (;) is optional; you should always use it to terminate a statement. For example, the following declares a variable and shows it to the console:

let message = "Welcome to JavaScript";

console.log(message);

Code language: JavaScript (javascript)

**Blocks**

A block is a sequence of zero or more simple statements. A block is delimited by a pair of curly brackets {}. For example

if (window.localStorage) {

console.log('The local storage is supported');

}

## Control flow Statements

* [if](https://www.javascripttutorial.net/javascript-if/) – show you how use the if statement to execute a block if a condition is true.
* [if…else](https://www.javascripttutorial.net/javascript-if-else/) – learn how to execute a block of code based on a specified condition.
* [if…else…if](https://www.javascripttutorial.net/javascript-if-else-if/) – check multiple conditions and execute a block.
* [Ternary operators](https://www.javascripttutorial.net/javascript-ternary-operator/) – show you how to make a shortcut for the if statement ( ?:).
* [switch](https://www.javascripttutorial.net/javascript-switch-case/) – show you how to replace multiple if statements when comparing a value with multiple variants by using the switch statement.
* [while](https://www.javascripttutorial.net/javascript-while-loop/) – learn how to perform a pre-test loop that repeatedly executes a block of code as long as a specified condition is true.
* [do…while](https://www.javascripttutorial.net/javascript-do-while/) – show you how to carry a post-test loop that executes a block of code repeatedly until a specified condition is false.
* [for loop](https://www.javascripttutorial.net/javascript-for-loop/) – learn how to repeatedly execute a block of code based on various options.
* [break](https://www.javascripttutorial.net/javascript-break/) – learn how to prematurely terminate a loop.
* [continue](https://www.javascripttutorial.net/javascript-continue/) – show you how to skip the current iteration of a loop and jump to the next one.
* [Comma operator](https://www.javascripttutorial.net/javascript-comma-operator/) – guide you on how to use the comma operator in a for loop to update multiple variables at once.

## Operators

* [Arithmetic operators](https://www.javascripttutorial.net/javascript-arithmetic-operators/) – introduce to you the arithmetic operators including addition (+), subtraction (-), multiplication (\*), and division (/).
* [Remainder operator](https://www.javascripttutorial.net/javascript-remainder-operator/) – show you how to use the remainder operator (%) to get the remainder left over when one value is divided by another value.
* [Assignment operators](https://www.javascripttutorial.net/javascript-assignment-operators/) – guide you on how to use assignment operators (=) to assign a value or an expression to a variable.
* [Unary operators](https://www.javascripttutorial.net/javascript-unary-operators/) – learn how to use unary operators.
* [Comparison operators](https://www.javascripttutorial.net/javascript-comparison-operators/) – show you how to use comparison operators to compare two values.
* [Logical operators](https://www.javascripttutorial.net/javascript-logical-operators/) – learn how to use the logical operators: NOT (!), AND (&&), and OR (||).
* [Logical assignment operators](https://www.javascripttutorial.net/es-next/javascript-logical-assignment-operators/) – introduce to you the logical assignment operators, including ||=, &&=, and ??=
* [Nullish coalescing operator](https://www.javascripttutorial.net/es-next/javascript-nullish-coalescing-operator/) (??) – accept two values and return the second value if the first one is null or undefined.
* [Exponentiation operator](https://www.javascripttutorial.net/es-next/javascript-exponentiation-operator/) – introduce you to the exponentiation operator (\*\*) that calculates a base to the exponent power, which is similar to Math.pow() method.

## Classes

* [Class](https://www.javascripttutorial.net/es6/javascript-class/) – introduce you to the ES6 class syntax and how to declare a class.
* [Getters and Setters](https://www.javascripttutorial.net/es6/javascript-getters-and-setters/) – define the getters and setters for a class using the get and set keywords.
* [Class Expression](https://www.javascripttutorial.net/es6/javascript-class-expressions/) – learn an alternative way to define a new class using a class expression.
* [Computed property](https://www.javascripttutorial.net/es6/javascript-computed-property/) – explain the computed property and its practical application.
* [Inheritance](https://www.javascripttutorial.net/es6/javascript-inheritance/) – show you how to extend a class using the extends and super keywords.
* [new.target](https://www.javascripttutorial.net/es6/javascript-new-target/) – introduce you to the new.target metaproperty.
* [Static methods](https://www.javascripttutorial.net/es6/javascript-static-method/) – guide you on how to define methods associated with a class, not instances of that class.
* [Static Properties](https://www.javascripttutorial.net/es6/javascript-static-properties/) – show you how to define static properties shared by all instances of a class.
* [Private Fields](https://www.javascripttutorial.net/javascript-private-fields/) – learn how to define private fields in a class.
* [Private Methods](https://www.javascripttutorial.net/javascript-private-methods/) – show you how to define private methods in a class.

## Collections

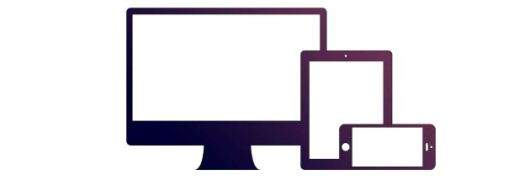
* [Map](https://www.javascripttutorial.net/es6/javascript-map/) – introduce you to the Map type that holds a collection of key-value pairs.
* [Set](https://www.javascripttutorial.net/es6/javascript-set/) – learn how to use the Set type that holds a collection of unique values.

## Why to Learn Bootstrap?

* **Mobile first approach** − Bootstrap 3, framework consists of Mobile first styles throughout the entire library instead them of in separate files.
* **Browser Support** − It is supported by all popular browsers.



* **Easy to get started** − With just the knowledge of HTML and CSS anyone can get started with Bootstrap. Also the Bootstrap official site has a good documentation.
* **Responsive design** − Bootstrap's responsive CSS adjusts to Desktops, Tablets and Mobiles. More about the responsive design is in the chapter [Bootstrap Responsive Design.](https://www.tutorialspoint.com/bootstrap/bootstrap_responsive_utilities.htm)



* Provides a clean and uniform solution for building an interface for developers.
* It contains beautiful and functional built-in components which are easy to customize.
* It also provides web based customization.
* And best of all it is an open source.

**BootStrap:**

Just to give you a little excitement about Bootstrap, I'm going to give you a small Bootstrap Program. You can edit and try running this code using Edit & Run option.

<!DOCTYPE html>

<html lang="en">

<head>

<title>Bootstrap 5 Example</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="container-fluid p-5 bg-primary text-white text-center">

<h1>Bootstrap Demo Page</h1>

<p>Resize this page to see the effect!</p>

</div>

<div class="container mt-5">

<div class="row">

<div class="col-sm-6">

<h3>Column 1</h3>

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit...</p>

<p>Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris...</p>

</div>

<div class="col-sm-6">

<h3>Column 2</h3>

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit...</p>

<p>Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris...</p>

</div>

</div>

</div>

</body>

</html>

## Applications of Bootstrap

* Scaffolding − Bootstrap provides a basic structure with Grid System, link styles, and background. This is is covered in detail in the section Bootstrap Basic Structure
* CSS − Bootstrap comes with the feature of global CSS settings, fundamental HTML elements styled and enhanced with extensible classes, and an advanced grid system. This is covered in detail in the section Bootstrap with CSS.
* Components − Bootstrap contains over a dozen reusable components built to provide iconography, dropdowns, navigation, alerts, pop-overs, and much more. This is covered in detail in the section Layout Components.
* JavaScript Plugins − Bootstrap contains over a dozen custom jQuery plugins. You can easily include them all, or one by one. This is covered in details in the section Bootstrap Plugins.
* Customize − You can customize Bootstrap's components, LESS variables, and jQuery plugins to get your very own version.

## Working of Bootstrap Grid System

Grid systems are used for creating page layouts through a series of rows and columns that house your content. Here's how the Bootstrap grid system works −

* Rows must be placed within a .container class for proper alignment and padding.
* Use rows to create horizontal groups of columns.
* Content should be placed within the columns, and only columns may be the immediate children of rows.
* Predefined grid classes like .row and .col-xs-4 are available for quickly making grid layouts. LESS mixins can also be used for more semantic layouts.
* Columns create gutters (gaps between column content) via padding. That padding is offset in rows for the first and the last column via negative margin on .rows.
* Grid columns are created by specifying the number of twelve available columns you wish to span. For example, three equal columns would use three .col-xs-4.

## Media Queries

Media query is a really fancy term for "conditional CSS rule". It simply applies some CSS, based on certain conditions set forth. If those conditions are met, the style is applied.

Media Queries in Bootstrap allow you to move, show and hide content based on the viewport size. Following media queries are used in LESS files to create the key breakpoints in the Bootstrap grid system.

/\* Extra small devices (phones, less than 768px) \*/

/\* No media query since this is the default in Bootstrap \*/

/\* Small devices (tablets, 768px and up) \*/

@media (min-width: @screen-sm-min) { ... }

/\* Medium devices (desktops, 992px and up) \*/

@media (min-width: @screen-md-min) { ... }

/\* Large devices (large desktops, 1200px and up) \*/

@media (min-width: @screen-lg-min) { ... }

Occasionally these are expanded to include a max-width to limit CSS to a narrower set of devices.

@media (max-width: @screen-xs-max) { ... }

@media (min-width: @screen-sm-min) and (max-width: @screen-sm-max) { ... }

@media (min-width: @screen-md-min) and (max-width: @screen-md-max) { ... }

@media (min-width: @screen-lg-min) { ... }

Media queries have two parts, a device specification and then a size rule. In the above case, the following rule is set −

Let us consider this line −

@media (min-width: @screen-sm-min) and (max-width: @screen-sm-max) { ... }

For all devices no matter what kind with *min-width: @screen-sm-min* if the width of the screen gets smaller than *@screen-sm-max*, *then do something*.

**BACK END**

**Firebase:**

-Authentication: Utilize Firebase Authentication to handle user authentication and secure access to backend services.

- Real-time Database: Utilize Firebase Realtime Database to store and retrieve data in real-time, such as user profiles, PG listings, and booking details.

- Cloud Firestore: Use Firebase Cloud Firestore, a NoSQL document database, for structured data storage and querying capabilities for features like user reviews and ratings.

- Cloud Functions: Implement Cloud Functions for Firebase to run server-side code and perform backend logic, such as sending notifications, handling booking requests, and updating data.

- Storage: Utilize Firebase Storage for storing and serving user-uploaded images or other media associated with PG listings.

- Push Notifications: Utilize Firebase Cloud Messaging (FCM) to send push notifications to users, such as updates on booking status or new PG listings matching user preferences.

- Security Rules: Configure Firebase Security Rules to define granular access control and protect sensitive data, ensuring that only authorized users can perform specific actions.

- Analytics: Utilize Firebase Analytics to gain insights into user behavior, monitor usage patterns, and make data-driven decisions to improve the platform.

- Crash Reporting: Implement Firebase Crashlytics to track and report application crashes, helping identify and fix bugs promptly.

- Performance Monitoring: Utilize Firebase Performance Monitoring to monitor backend performance metrics, identify bottlenecks, and optimize API response times.

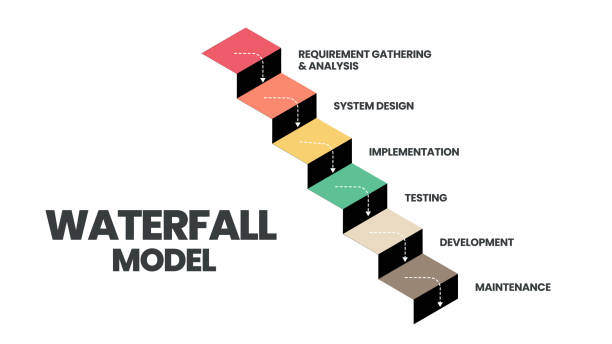
- Integration with other Firebase services: Leverage additional Firebase services like Firebase Cloud Storage, Firebase Remote Config, or Firebase Test Lab for additional functionalities or testing needs as required by the backend system.

**5.SOFTWARE PROCESS MODEL:**

**Waterfall Model**

The waterfall model is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Requirement Analysis, Design, Coding, Testing, Deployment, and Maintenance .

Following is a diagrammatic representation of different phases of waterfall model.



The sequential phases in Waterfall model are:

* **Requirement Gathering and analysis**

All possible requirements of the system to be developed are captured in this phase. Requirements are set of functionalities and constraints that the end-user (who will be using the system) expects from the system. The requirements are gathered from the end-user by consultation, these requirements are analyzed for their validity and the possibility of incorporating the requirements in the system to be development is also studied. Finally all requirements documented in a requirement specification doc.

* **System Design**

Before a starting for actual coding, it is highly important to understand what we are going to create and what it should look like? The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The system design specifications serve as input for the next phase of the model.

* **Implementation**

With inputs from system design, the work is divided in modules/units and actual coding is started. The system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing. Unit testing mainly verifies if the modules/units meet their specifications.

* **Integration and Testing**

All the units developed in the implementation phase are integrated into a system after testing of each unit. These units are integrated into a complete system during Integration phase and tested to check if all modules/units coordinate between each other and the system as a whole behaves as per the specifications. Post integration the entire system is tested for any faults and failures.

* **Maintenance**

This phase of "The Waterfall Model" is virtually never ending phase. There are some issues which come up in the client environment. Not all the problems come in picture directly but they arise time to time and needs to be solved. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap. Waterfall model is the earliest SDLC approach that was used for software development.

**Waterfall Model Application**

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

* Requirements are very well documented, clear and fixed.
* Product definition is stable.
* Technology is understood and is not dynamic.
* There are no ambiguous requirements.
* Ample resources with required expertise are available to support the product.

## Waterfall Model Pros & Cons:

## Advantage

The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

### Disadvantage

The disadvantage of waterfall development is that it does not allow for much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage. Not suitable for the projects where requirements are at a moderate to high risk of changing. So risk and uncertainty is high with this process model.

Why we use waterfall model?

As it is a major project and being a beginner, we already have the requirements for our ongoing project. Waterfall model is considered to be of downward approach and we don’t have to look up to the previous level that frequently, it’s beneficial for our project to complete it in a timely manner.Thus if we want to modify anything within our project after deployment, we can start from the initial phase. Thus it does not freeze the possibility for any kind of change.

**6. DESIGN**

**6.1 SYSTEM DESIGN**

The most creative and challenging phase of SDLC is system design. The term design describes a final system and the process by which it is developed. It includes construction of programs and program testing.

The purpose of the design phase is to plan a solution of the problem specifies by the requirements document. This phase is the first step in the moving from the problem domain to the solution domain. Starting with what is needed; design takes us towards how to satisfy the needs. The design of the system is perhaps the most critical factor affecting the quality of the software. It has major impact on the later phase, particularly testing and maintenance. The output of this phase is the design document. This document is similar to blueprint or plan for the solution and is used later during implementation, testing and maintenance.

A systematic method has to achieve the beneficial result at the end. It includes starting with average idea and developing it into a series of steps. The series of steps for successful system development are given below:

* Study problem completely because first of all we should know the goal, which he has to achieve.
* We should see what kind of output we require and what kind of input we give so we can get the desired output from system output from system. It is very challenging step of system development.
* According to the output requirement of system the strength of various databases should be design.
* Next, we should know what kind of program we should develop, which will lead us to reach final goal.
* Then we write this individual program, which later on joining solve problem.
* Then we test these programs and make necessary correction in them to achieve target of program.
* At last combining all these problems in the forms of a bar in the menu of windows, this will complete software package for general insurance.

The three main objectives which the designer has to bear in mind are:-

1. How fast the design will be do the users work given particular hardware resources.
2. The extent to which the design is secure against the human errors and machine malfunctions.
3. The ease with which the design allows the system to be changed.

To meet these objectives analyst and programmers use a top-down and bottom-up design.

* TOP – DOWN DESIGN

It is also known as system design, aims to identify the modules that should be in a system. It starts with large picture and move to the details. The analyst and team members look at major functions that the system must provide and break these down into smaller and smaller activities.

* BOTTOM – UP APPROACH

It is also known as detailed design. It starts with details and then moves to the big picture. This approach is appropriate when users have specific requirements for output.

**6.2 DFD: Data Flow Diagram**

Data Flow Diagrams were first developed by Larry Constantine as a way of expressing system requirements in a graphical form. DFD is also known as bubble chart and has a purpose of clarifying system requirements and identifying major transformations and will become the program in the system design.

Data Flow Diagramming is a means of representing a system at any level of detail with a graphic network of symbols showing data flows, data stores, data processes, and data sources/destinations.

Purpose**:**

The purpose of data flow diagrams is to provide a semantic bridge between users and systems developers.

The diagrams are:

* Graphical, eliminating thousands of words.
* Logical representations, modeling WHAT a system does, rather than physical models showing HOW it does it.
* hierarchical, showing systems at any level of detail and
* Allowing user understanding and reviewing.

DFD Symbols are as follows:

* The External Entity symbol represents sources of data to the system or destinations of data from the system.

* The Data Flow symbol represents movement of data.

* The Data Store symbol represents data that is not moving (delayed data at rest).

* The Process symbol represents an activity that transforms or manipulates the data.

**Level 0 DFD:**

**Context Level Diagram**

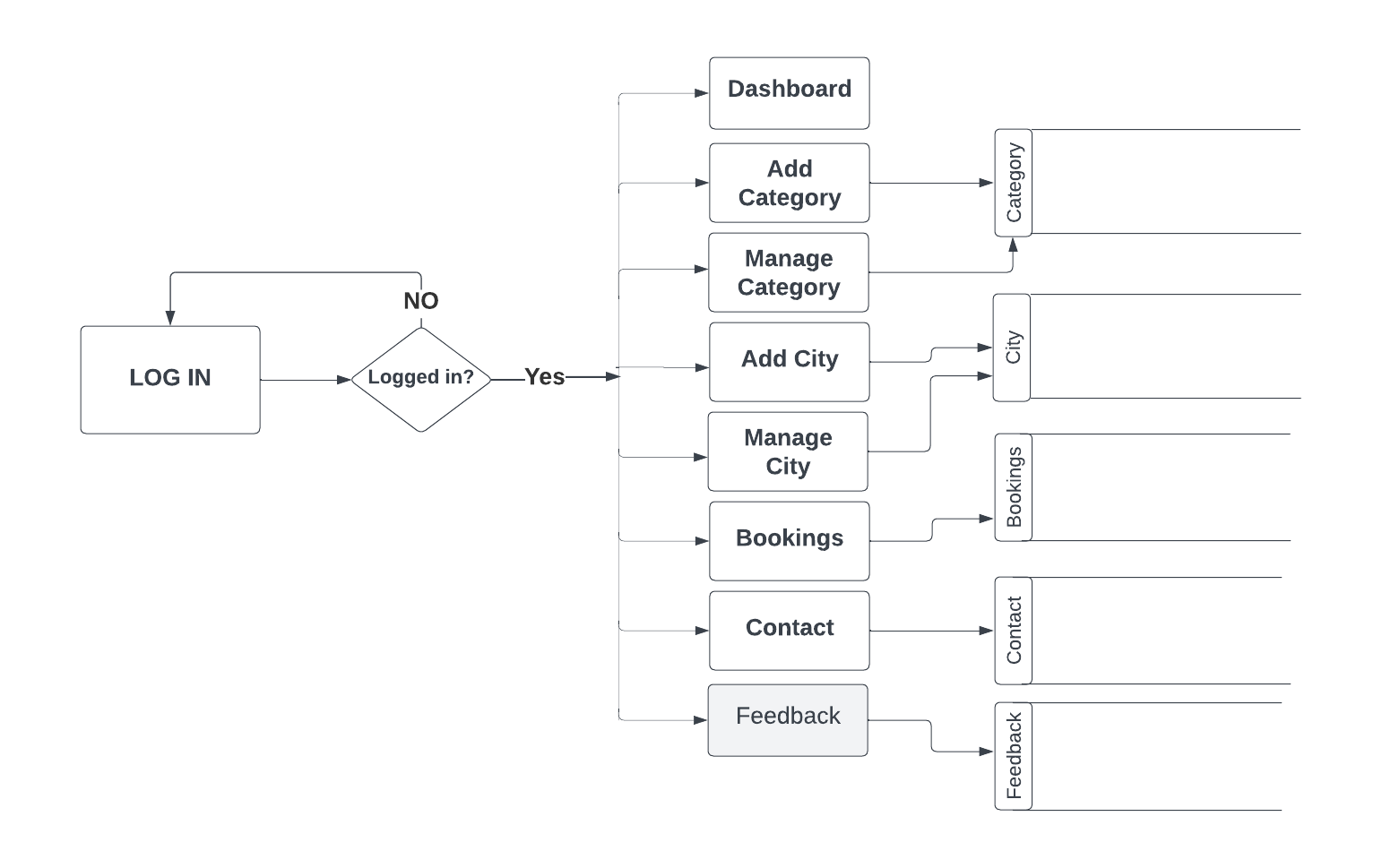
User

Admin

Here User and Admin interact with the system for different purposes. Database contains all information which user need.

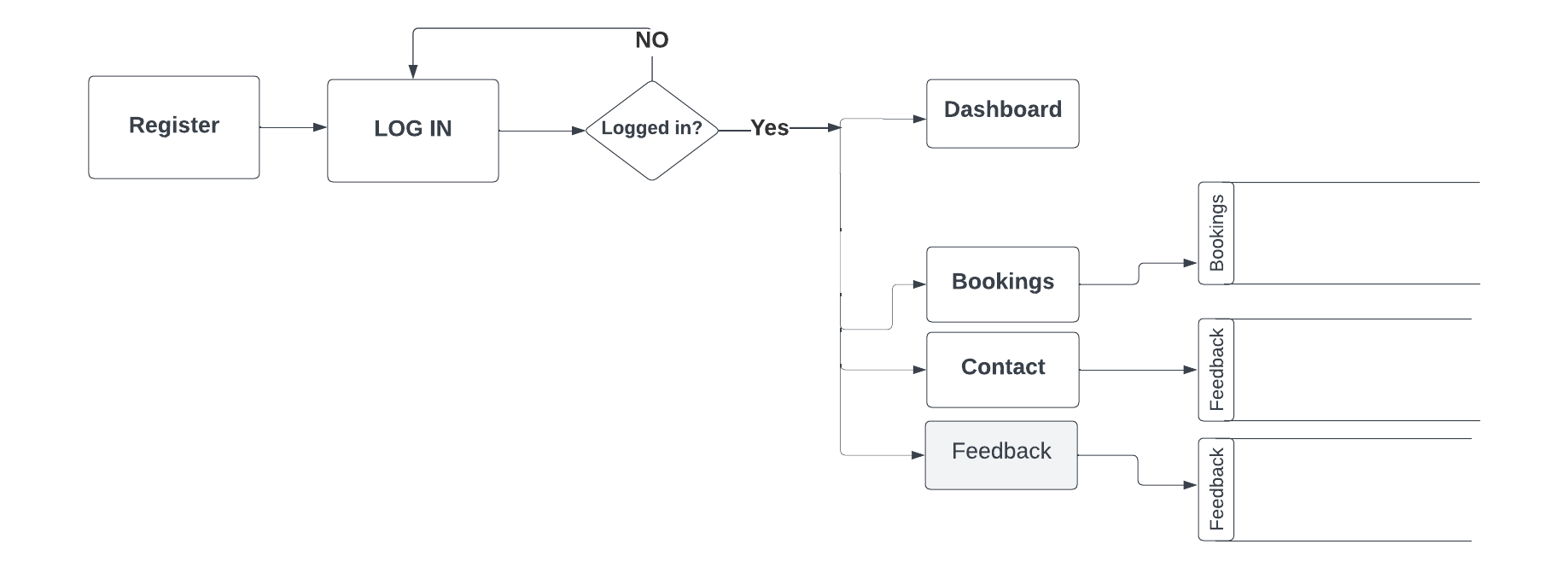
**Level 1 DFD:**

**DFD For Admin**



**Level 1 DFD:**

**DFD For User**



**6.4Database Tables:**

***7.* CODING:**

**Header Page**

<!--

Author: W3layouts

Author URL: http://w3layouts.com

License: Creative Commons Attribution 3.0 Unported

License URL: http://creativecommons.org/licenses/by/3.0/

-->

<!DOCTYPE html>

<html lang="en">

<head>

<title>Official Corporate Category Flat Bootstrap Responsive Website Template | Home : W3layouts</title>

<!-- Meta tag Keywords -->

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta charset="utf-8">

<meta name="keywords" content="Official Responsive web template, Bootstrap Web Templates, Flat Web Templates, Android Compatible web template,

Smartphone Compatible web template, free webdesigns for Nokia, Samsung, LG, SonyEricsson, Motorola web design" />

<script>

addEventListener("load", function () {

setTimeout(hideURLbar, 0);

}, false);

functionhideURLbar() {

window.scrollTo(0, 1);

}

</script>

<!--// Meta tag Keywords -->

<!--css files -->

<link rel="stylesheet" href="css/bootstrap.css"><!-- Bootstrap-Core-CSS -->

<link rel="stylesheet" href="css/style.css" type="text/css" media="all" /><!-- Style-CSS -->

<link rel="stylesheet" href="css/font-awesome.min.css" type="text/css" media="all" />

<!-- Style-CSS -->

<!-- //css files -->

<!--web font-->

<link href="//fonts.googleapis.com/css?family=Work+Sans:100,200,300,400,500,600,700,800,900" rel="stylesheet">

<!--//web font-->

</head>

<body>

<!-- header -->

<header>

<div class="container">

<div class="header d-lg-flex justify-content-between align-items-center">

<div class="header-agile">

<h1>

<a class="navbar-brand logo" href="index.php">

<span class="fa fa-folder-open-o"></span>Official

</a>

</h1>

</div>

<div class="nav\_w3ls">

<nav>

<label for="drop" class="toggle mt-lg-0 mt-2"><span class="fa fa-bars" aria-hidden="true"></span></label>

<input type="checkbox" id="drop" />

<ul class="menu">

<li class="mr-lg-3 mr-2"><a href="index.php">Home</a></li>

<li class="mr-lg-3 mr-2"><a href="about.php">About</a></li>

<li class="mr-lg-3 mr-2"><a href="services.php">Services</a></li>

<li class="mr-lg-3 mr-2 p-0">

<!-- First Tier Drop Down -->

<label for="drop-2" class="toggle">Login <span class="fa fa-angle-down" aria-hidden="true"></span></label>

<a href="#">Login <span class="fa fa-angle-down" aria-hidden="true"></span></a>

<input type="checkbox" id="drop-2"/>

<ul class="inner-dropdown">

<li><a href="user/index.php">USER</a></li>

<li><a href="staff/index.php">Staff</a></li>

<li><a href="admin/index.php">Admin</a></li>

</ul>

</li>

<li><a href="projects.php">Projects</a></li>

</ul>

</nav>

</div>

<div class="buttons mt-lg-0 mt-2">

<a href="contact.php">Contact Us </a>

</div>

</div>

</div>

</header>

<!-- //header -->

<!-- banner -->

<div class="banner\_w3lspvt" id="home">

<div class="container">

<div class="row banner-tops-style">

<div class="col-lg-8 style-banner">

<h3 class="text-wh"> Creative Agencies <br> Also tends to Cover</h3>

<p class="text-li mt-4">Praesent at molestienibh, eulaoreetmassa. Nam tristiquetortorrisus, vitae ornarediamtincidunt vitae. In in porta arcu. Integer non convallis mauris. Quisque at nunc at nibhdapibusrutrumveleumetus. Nullamlaoreetvulputatetortor.</p>

<a href="about.php" class="btn button-style mt-sm-5 mt-4">Know About Us</a>

<a href="#" class="btn call mt-sm-5 mt-4"><span class="fa fa-phone" aria-hidden="true"></span> Call: +012 898 909 2317</a>

</div>

</div>

</div>

</div>

<!-- //banner -->

**Footer Page:**

<!-- brands -->

<section class="brands py-5" id="brands">

<div class="container py-lg-0">

<div class="row text-center">

<div class="col-sm-2 col-3">

<a href="#"><span class="fa fa-connectdevelop" aria-hidden="true"></span></a>

</div>

<div class="col-sm-2 col-3">

<a href="#"><span class="fa fa-empire" aria-hidden="true"></span></a>

</div>

<div class="col-sm-2 col-3">

<a href="#"><span class="fa fa-ioxhost" aria-hidden="true"></span></a>

</div>

<div class="col-sm-2 col-3">

<a href="#"><span class="fa fa-first-order" aria-hidden="true"></span></a>

</div>

<div class="col-sm-2 col-3 mt-sm-0 mt-4">

<a href="#"><span class="fa fa-joomla" aria-hidden="true"></span></a>

</div>

<div class="col-sm-2 col-3 mt-sm-0 mt-4">

<a href="#"><span class="fa fa-dropbox" aria-hidden="true"></span></a>

</div>

</div>

</div>

</section>

<!-- brands -->

<!-- footer -->

<footer class="py-5">

<div class="container py-md-3">

<div class="row footer-grids">

<div class="col-md-4">

<div class="footer-grid left">

<h2 class="logo"><a href="index.php"><span class="fa fa-folder-open-o"></span>Official</a></h2>

</div>

</div>

<div class="col-md-4 middle">

<p class="btn call"><span class="fa fa-phone"></span>Call: +012 898 909 2317</p>

</div>

<div class="col-md-4 right">

<!-- to top -->

<div class="top-icon">

<a href="#home" class="move-top text-center"><span class="fa fa-angle-up mb-3" aria-hidden="true"></span>To Top</a>

</div>

<!-- //to top -->

</div>

</div>

<div class="middle mt-3">

<p>© 2019 Official. All Rights Reserved | Design by <a href="http://www.W3Layouts.com" target="\_blank">W3Layouts</a></p>

<ul class="social mt-4">

<li><a href="#"><span class="fa fa-facebookicon\_facebook"></span></a></li>

<li><a href="#"><span class="fa fa-twitter icon\_twitter"></span></a></li>

<li><a href="#"><span class="fa fa-google-plus icon\_google-plus"></span></a></li>

<li><a href="#"><span class="fa fa-linkedinicon\_linkedin"></span></a></li>

<li><a href="#"><span class="fa fa-dribbbleicon\_dribbble"></span></a></li>

</ul>

</div>

</div>

</footer>

<!-- //footer -->

</body>

</html>

**Login Page:**

<?php

include 'header1.php';

if(isset($\_SESSION["studentsession"]))

{

echo "<script>window.location.assign('welcome.php')</script>";

}

?>

<div class="container">

<div class="row">

<div class="col-md-4"></div>

<div class="col-md-4">

<center><h1>Student Login</h1></center>

<form action="check.html" method="post">

<label>Rollno</label>

<input type="text" name="studentsession" class="form-control" placeholder="Enter Rollno">

<label>Password</label>

<input type="password" name="password" class="form-control" placeholder="Enter Password">

<center><input type="submit" name="submit" class="btnbtn-outline-primary btn-submit"></center>

<center><a href="forget.html">Forget Password</a></center>

</form>

</div>

</div>

</div>

**8. SNAPSHOTS**

Fig 8.1 Admin Login

Fig 8.2 Admin Dashboard

Fig 8.3 Manage Faculty

Fig 8.4 Manage Students

Fig 8.5 Manage Courses

Fig 8.6 Manage Subjects

Fig 8.7Contact

Fig 8.8 Manage Admin Profile

-

Fig 8.9 Faculty Login

Fig 8.10 Faculty Dashboard

Fig 8.11 Faculty Profile

Fig 8.12 Student Login

Fig 8.13 Student Dashboard

Fig 8.14 View Assignment

Fig 8.15 View Marks

Fig 8.16 View Internal Assessment

Fig 8.17 Student Profile

**9. BIBLIOGRAPHY**

**Websites:**

* [www.google.com](http://www.google.com)
* [www.tutorialspoint.com](http://www.tutorialspoint.com)
* [www.javatpoint.com](http://www.javatpoint.com)
* [www.npmjs.com](http://www.npmjs.com)
* www.bootstapmade.com