**ORACLE APEX APPLICATION DEVELOPMENT**

An Industrial Internship Report

*submitted by*

**Sudhanshu Singh**

**(1903600100053)**

*in partial fulfillment for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE AND ENGINEERING**



**GOEL INSTITUTE OF TECHNOLOGY & MANAGEMENT, LUCKNOW**

**Affiliated**

**A. P. J. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW, INDIA**

OCTOBER 2022

**DECLARATION BY THE CANDIDATE**

I hereby declare that the Industrial Internship report entitled “**ORACLE APEX APPLICATION DEVELOPMENT”** submitted by me to GOEL Institute of Technology, GOEL in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** is a record of bonafide industrial training undertaken by me under the supervision of **Mr. Pramod Dixit at the PD Workforce Developers and Mr. Shivam Shukla at Goel Institute of Technology and Management, Lucknow.** I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

**Vellore: Signature of the Candidate**

**Date: 07-oct-2022**



**GOEL INSTITUTE OF TECHNOLOGY & MANAGEMENT, LUCKNOW**

**Affiliated**

**A. P. J. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW, INDIA**

**BONAFIDE CERTIFICATE**

This is to certify that the Industrial Internship report entitled “**ORACLE APEX APPLICATION DEVELOPMENT”** submitted by **Sudhanshu Singh (1903600100053)** to GOEL Institute of Technology, Lucknow in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science And Engineering is a record of bonafide Industrial Internship undertaken by him under my supervision. The training fulfills the requirements as per the regulations of this Institute and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Mr. Shivam Shukla

**SUPERVISOR**

**(**Assistant professor**)**

Date:Date:

**Internal Examiner (s) External Examiner (s)**

**CERTIFICATE BY TRAINING OFFICER**



**ACKNOWLEDGEMENT**

I would like to express my gratitude to my Goel Faculty supervisor **Mr. Shivam Shukla** of **Computer Science and Engineering Department ,Goel Institute of Technology And Management, Lucknow** for his encourage support and guidance in carrying out the project.

I would like to thanks to my internal guide **Mr. Shivam Shukla** for his Valuable Guidance and Suggestions during this project which helped me a lot to do the required things in my project.

I am Also very Thankfull to My HOD Sir **Mr. Brijesh Pandey** and My Co-ordinator Sir **Mr. Yogendra Pratap singh**. Thankfull for my faculty **Mr. Peeyush Pathak, Miss Samiksha singh, Durgesh Tiwari, Mr. Nitin Kumar Rao, Ajay Kumar** for their proper motivation for project work.

With great pleasure, I would like to express my sincere gratitude to **Mr. Pramod dixit, Director, PD Workforce Developer, Lucknow**.

I want to give my heartiest thanks to my parents whose principle and morale helped me to rise to this position. I would also like to thank my all friends who provided useful suggestion.Last but not least I would like to thanks for God for all he has given me till today.

**Place:** Lucknow

**(Sudhanshu Singh)**

**Date:** 07-oct-2022

**ABSTRACT**

Oracle APEX is an amazing low-code development platform in which you can build robust web applications. Not only it provides an environment where you can rapidly develop data-centric web applications, it also allows end users to interact with their data via tools like interactive report, interactive grid, faceted search, different types of charts and more.

The ultimate objective of Oracle APEX is to building web applications by iteratively developing the sample database application from scratch. The short list below presents some main features of Oracle APEX.

Browser-based online application development. Rapid web application development for desktops, laptops, tablets, and latest smartphones. Create comprehensive applications declaratively without writing tons of code Tweak application pages using Page Designer. Create applications with the help of wizards. Create custom application pages by adding components manually. Use same interface and code to develop applications for a wide array of devices. Present data using a variety of eye ‐ catching charts Produce highly formatted PDF reports, including invoices, grouped reports, and pivot tables Implement. APEX’s built-in security module.

“Inventory Management System” is the Project which is developed through Oracle Apex at Front-End in which Oracle Apex tools and HTML,CSS are Used & at Back-End oracle 21c is Used. Oracle 21c is a cloud version of Database in which SQL and PL/SQL Languages are used.

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **TITLE** | **PAGE NO.** | **SIGNATURE** |
| **1.** | Introduction Of Oracle Apex | 8-26 |  |
| **2.** | Study of the existing System in Industry | 27-31 |  |
| **3.** | Analysis/Proposed Work | 32-34 |  |
| **4.** | Implementation | 35-46 |  |
| **5.** | Result and Conclusion/Discussion | 47-49 |  |

**CHAPTER-1**

**INTRODUCTION OF ORACLE APEX**

If you are interested in developing professional web applications rapidly, then you have chosen the right track. Oracle APEX is a rapid application development (RAD) tool that runs inside an Oracle database instance and comes as a free option with Oracle database. Using this unique tool you can develop and deploy fast and secure professional web applications. The only requirements are a web browser and a little SQL and PL/SQL experience.

Oracle APEX provides a declarative programming environment, which means that no code is generated nor compiled during development. You just interact through wizards and property editor to build web applications on existing database schemas. Reports and charts are defined with simple SQL queries, so some knowledge of SQL is very helpful. If you want to create more robust applications, then you can add procedural logic by writing PL/SQL code. Oracle APEX is a declarative tool and has a vast collection of pre-defined wizards, HTML objects, database handling utilities, page rendering and submission processes, navigation and branching options, and more. You can use all these options to build your database-centric web applications comprising web pages carrying forms, reports, charts, and so on with their layouts and business logic. The APEX engine translates it all into an HTML code for the client side and SQL and PL/SQL code for the server side. If you do not get a solution from built-in options, you are allowed by Oracle APEX to create your own SQL and PL/SQL code for the server side and HTML, CSS, and JavaScript code for the client side.

**DEMANDS OF APEX:-**

Velocity in the demand for new applications and functionality rises as businesses grow. As a developer, you are expected to rapidly respond to these needs. Over the years, desktop database and spreadsheet tools have enormously contributed to data management due to the ease and user friendliness these applications extend to their users. Besides benefits, these applications have scalability and functionality limitations that not only results in dozens of different applications and data sources but also adds extra overhead in their maintenance. Because of these issues, organizations are unable to continue their standard practices, leaving mission-critical data at risk. These fragmented systems may also cause loss of business opportunities. Last but by no means least, significant amount of time and resource is required to put these data blocks together to get the desired information.

A news article of Times of India shows the demand and need of Apex now a days:-



**ADVANTAGES OF APEX:-**

**Low-Code**: A low-code platform in which enterprise apps are built 20x faster with 100x less code.

**Robust and Proven:** Oracle APEX is capable to produce a wide variety of apps for any industry – from the simplest app that is created from a spreadsheet file, to mission-critical apps which are used daily by tens of thousands of users. The elegant architecture of Oracle APEX has been used to power thousands of applications around the globe for years. Oracle APEX has a much lower barrier to entry for creating responsive and powerful business applications.

**Installation:** No installation of software is required on client machines – the only requirement is a supported browser.

**Central Management:** Being central, data and applications become a part of regular backup procedure.

**Secure:** Data and application access control, empowered by audit trail. Oracle APEX is designed to build web apps which are highly secure out of the box. In a world of ever-changing web standards, evolving security standards, and resourceful hackers, the focus on security means that your applications stay protected and remain state-of-the-art.

**Portable:** You can run Oracle APEX everywhere – on the Oracle Cloud, on-premises, or anywhere else there is an Oracle Database. And you can deploy your Oracle APEX applications across any environment with ease.

**Reporting:** Oracle APEX includes powerful self-service reporting features. You can easily add custom filters, sort, aggregate, pivot and chart your data, and even create reports which get emailed to you on a periodic basis.

**Apps for Enterprises:** In an enterprise setting, Oracle APEX provides a scalable and proven platform for applications which can scale across the enterprise. Oracle APEX includes native functionality to integrate with REST and SOAP Services in your organization and in the cloud.

**PREREQUISITES:-**In the Oracle Apex Application Development you have to know some Basics of Html, CSS, JavaScript, Oracle 21C.

**HTML5:**

**What is HTML?**

* The standard markup language for developing web pages is Hyper Text Markup Language (HTML).
* HTML is used to describe how a web page will appear in a browser’s window. • HTML provides simple mechanisms for formatting text, creating links & lists, inserting images, embedding audio & video, etc.
* HTML documents are written using HTML “tags” embedded in angular brackets.
* The current version is HTML 5.
* HTML documents must have the extension .html.

**Basic Structure of HTML page**:

<html>

<head>

<! -- meta-information about the HTML page. -->

</head>

<body>

<! -- contains text and other tags, which are rendered on the screen as specified. -->

</body>

</html>

**Main root:**

**<html> ... </html>**

The HTML <html> element represents the root (top-level element) of an HTML

document, so it is also referred to as the root element. All other elements must be

descendants of this element.

Example:

<!DOCTYPE html>

<html lang="en">

<head>...</head>

<body>...</body>

</html

**Document metadata:**

**<head> ... </head>**

The HTML element contains machine-readable information (metadata) about the document, like its title, scripts, and style sheets.

**<link>**

The HTML External Resource Link element (<link>) specifies relationships

between the current document and an external resource. This element is most

commonly used to link to stylesheets but is also used to establish site icons (both

"favicon" style icons and icons for the home screen and apps on mobile devices)

among other things.

**<meta>**

The HTML <meta> element represents metadata that cannot be represented by

other HTML meta-related elements, like <base>, <link>, <script>, <style> or

<title>

<style> ... </style>

The HTML <style> element contains style information for a document, or part of

a document.

**<title> ... </title>**

The HTML Title element (<title>) defines the document's title that is shown in a

browser's title bar or a page's tab.

**Example:**

<!DOCTYPE html>

<html lang="en">

<head>...</head>

<body>...</body>

</html>

**Sectioning root:**

**<body> ... </body>**

The HTML <body> Element represents the content of an HTML document.

There can be only one <body> element in a document.

**Example:**

<html>

<head>

<title>Document title</title>

</head>

<body>

<p>This is a paragraph</p>

</body>

</html>

**<footer> ... </footer>**

The HTML <footer> element represents a footer for its nearest sectioning content

or sectioning root element. A footer typically contains information about the

author of the section, copyright data, or links to related documents.

**<header> ... </header>**

The HTML <header> element represents introductory content, typically a group

of introductory or navigational aids. It may contain some heading elements but

also a logo, a search form, an author name, and other elements.

**<h1> to <h6>**

The HTML <h1>–<h6> elements represent six levels of section headings. <h1>

is the highest section level and <h6> is the lowest.

**Example:**

<address>

<a href="mailto:jim@rock.com">jim@rock.com</a><br>

<a href="tel:+13115552368">(311) 555-2368</a>

</address>

**Text content:**

**<dd> ... </dd>**

The HTML <dd> element provides the description, definition, or value for the

preceding term (<dt>) in a description list (<dl>).

**<div> ... </div>**

The HTML Content Division element (<div>) is the generic container for flow

content. It has no effect on the content or layout until styled using CSS.

**<dl> ... </dl>**

The HTML <dl> element represents a description list. The element encloses a list

of groups of terms (specified using the <dt> element) and descriptions (provided

by <dd> elements).

Common uses for this element are to implement a glossary or to display metadata

(a list of key-value pairs).

**<dt> ... </dt>**

The HTML <dt> element specifies a term in a description or definition list, and

as such must be used inside a <dl> element.

**<hr>**

The HTML <hr> element represents a thematic break between paragraph-level

elements: for example, a change of scene in a story, or a shift of topic within a

section.

**<li> ... </li>**

The HTML <li> element is used to represent an item in a list.

**<ol> ... </ol>**

The HTML <ol> element represents an ordered list of items, typically rendered

as a numbered list.

**<p> ... </p>**

The HTML <p> element represents a paragraph.

**<pre> ... </pre>**

The HTML <pre> element represents preformatted text which is to be presented

exactly as written in the HTML file.

**<ul> ... </ul>**

The HTML <ul> element represents an unordered list of items, typically rendered

as a bulleted list.

**Examples:**

<dl>

<dt>Denim (semigloss finish)</dt>

<dd>Ceiling</dd>

<dt>Denim (eggshell finish)</dt>

<dt>Evening Sky (eggshell finish)</dt>

<dd>Layered on the walls</dd>

</dl>

<figure>

<img src="/media/examples/elephant-660-480.jpg"

alt="Elephant at sunset">

<figcaption>An elephant at sunset</figcaption>

</figure>

<ol>

<li>Mix flour, baking powder, sugar, and salt.</li>

<li>In another bowl, mix eggs, milk, and oil.</li>

<li>Stir both mixtures together.</li>

<li>Fill muffin tray 3/4 full.</li>

<li>Bake for 20 minutes.</li>

</ol>

**Inline text semantics:**

**<a> ... </a>**

The HTML <a> element (or anchor element), with its href attribute, creates a

hyperlink to web pages, files, email addresses, locations in the same page, or

anything else a URL can address.

**<br>**

The HTML <br> element produces a line break in text (carriage-return). It is

useful for writing apoem or an address, where the division of lines is significant.

**<em> ... </em>**

The HTML <em> element marks text that has stress emphasis. The <em> element

can be nested, with each level of nesting indicating a greater degree of emphasis.

**<span> ... </span>**

The HTML <span> element is a generic inline container for phrasing content,

which does not inherently represent anything. It can be used to group elements

for styling purposes (using the class or id attributes), or because they share

attribute values, such as lang.

**<strong> ... </strong>**

The HTML Strong Importance Element (<strong>) indicates that its contents

have strong importance, seriousness, or urgency. Browsers typically render the

contents in bold type.

**Image and multimedia:**

**<audio> ... </audio>**

The HTML <audio> element is used to embed sound content in documents. It

may contain one or more audio sources, represented using the src attribute or the

<source> element: the browser will choose the most suitable one. It can also be

the destination for streamed media, using a Media Stream.

**<img>**

The HTML <img> element embeds an image into the document**.**

**<video> ... </video>**

The HTML Video element (<video>) embeds a media player which supports

video playback into the document. You can use <video> for audio content as well,

but the <audio> element may provide a more appropriate user experience.

**Examples:**

<img class="fit-picture"

src="/media/examples/grapefruit-slice-332-332.jpg"

alt="Grapefruit slice atop a pile of other slices">

<figure>

<figcaption>Listen to the T-Rex:</figcaption>

<audio

controls

src="/media/examples/t-rex-roar.mp3">

Your browser does not support the

<code>audio</code> element.

</audio>

</figure>

<video controls width="250"

src="/media/examples/friday.mp4">

<track default kind="captions"

srclang="en"

src="/media/examples/friday.vtt"/>

Sorry, your browser doesn't support embedded videos.

</video>

**Table content:**

**<caption> ... </caption>**

The HTML Table Caption element (<caption>) specifies the caption (or title) of a table, and if used is always the first child of a <table>.

**<table> ... </table>**

The HTML <table> element represents tabular data — that is, information

presented in a two-dimensional table comprised of rows and columns of cells

containing data.

**<tbody> ... </tbody>**

The HTML Table Body element (<tbody>) encapsulates a set of table rows (<tr>

elements), indicating that they comprise the body of the table (<table>).

**<td> ... </td>**

The HTML <td> element defines a cell of a table that contains data. It participates

in the table model.

**<th> ... </th>**

The HTML <th> element defines a cell as a header of a group of table cells. The

exact nature of this group is defined by the scope and headers attributes.

**<thead> ... </thead>**

The HTML <thead> element defines a set of rows defining the head of the

columns of the table.

**<tr> ... </tr>**

The HTML <tr> element defines a row of cells in a table. The row's cells can then

be established using a mix of <td> (data cell) and <th> (header cell) elements.

**<tfoot> ... </tfoot>**

The HTML <tfoot> element defines a set of rows summarizing the columns of

the table.

**Example:**

<table>

<thead>

<tr>

<th colspan="2">The table header</th>

</tr>

</thead>

<tbody>

<tr>

<td>The table body</td>

<td>with two columns</td>

</tr>

</tbody>

</table>

**Forms:**

**<button> ... </button>**

The HTML <button> element represents a clickable button, which can be used in

forms or anywhere in a document that needs simple, standard button

functionality.

**<datalist> ... </datalist>**

The HTML <datalist> element contains a set of <option> elements that represent

the values available for other controls.

**<fieldset> ... </fieldset>**

The HTML <fieldset> element is used to group several controls as well as labels

(<label>) within a web form.

**<form> ... </form>**

The HTML <form> element represents a document section that contains

interactive controls for submitting information to a web server.

**<input>**

The HTML <input> element is used to create interactive controls for web-based

forms in order to accept data from the user; a wide variety of types of input data

and control widgets are available, depending on the device and user agent.

**<label> ... </label>**

The HTML <label> element represents a caption for an item in a user interface.

**<option> ... </option>**

The HTML <option> element is used to define an item contained in a <select>,

an <optgroup>, or a <datalist> element. As such, <option> can represent menu

items in popups and other lists of items in an HTML document.

**<progress> ... </progress>**

The HTML <progress> element displays an indicator showing the completion

progress of at the task, typically displayed as a progress bar.

**<select> ... </select>**

The HTML <select> element represents a control that provides a menu of options.

**<textarea> ... </textarea>**

The HTML <textarea> element represents a multi-line plain-text editing control,

useful when you want to allow users to enter a sizeable amount of free-form text,

for example, a comment on a review or feedback form.

**Example:**

<form action="" method="get" class="form-example">

<div class="form-example">

<label for="name">Enter your name: </label>

<input type="text" name="name" id="name" required>

</div>

<div class="form-example">

<label for="email">Enter your email: </label>

<input type="email" name="email" id="email" required>

</div>

<div class="form-example">

<input type="submit" value="Subscribe!">

</div>

</form>

**CSS3:**

• Cascading Style Sheet (CSS) is a style sheet language that specifies how to incorporate style information in a style sheet.

• The term ‘cascading’ indicates that several style sheets can be blended to present a document on the browser’s screen.

• A style sheet is a document that contains style information about one or more documents written in markup languages.

• CSS enables us to control rendering of styles such as fonts, color, spacing, margins, and other aspects of document style.

There are four ways to specify style information in a document:

External Style Sheets is specified using the HTML <link> tag in the head section.

<link rel= “stylesheet” type= “text/css” href= “filename.css” >

Embedded Style Sheets

<style>

p {

color: green;

}

</style>

Imported Style Sheets allows importing a style sheet from another style sheet.

<style>

@ import url(“filename.css”);

</style>

Inline Style Sheets are incorporated directly into the HTML tags.

<p style= “color: red”>Hello World! < /p>

**JavaScript:**

* It is a scripting language and supports client-side scripting.
* It is an interpreted language.
* It was developed by Netscape Corporation.
* Initially it was known as “Live Script” It was designed to add interactivity to HTML pages.
* It also supports server-side scripting, separately known as Livewire.
* Currently supported by most of the web browser Microsoft Internet Explorer, Google Chrome, Mozilla Firefox, etc.

**Limitations of JavaScript:**

* Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
* JavaScript cannot be used for networking applications because there is no such support available.
* JavaScript doesn't have any multithreading or multiprocessor capabilities. Strength of Java Script/ What JavaScript can do?
* Validate data: It can be used to validate form data before it is submitted to a server. This saves the server from extra processing and network congestion.
* Put dynamic text into a HTML page: A JavaScript statement can write a variable text into an HTML page.
* Detect the visitor's browser: It can be used to detect the visitor's browser, and depending on the browser load another page specifically designed for that browser.
* React to events: It can be set to execute when something happens, like when a page has finished loading or when a user clicks on a HTML element.
* Create cookies: It can be used to store and retrieve information on the visitor's Page 21 of 47 computer.

**Syntax:**

**< SCRIPT language="javascript" type="text/javascript">...</SCRIPT>**

The script tag takes two attributes:

**Language:** This attribute specifies which scripting language we are using. Typically, its value will be JavaScript. Although recent versions of HTML have phased out the use of this attribute.

**Type:** This attribute is what is now recommended to indicate the scripting language in use and its value should be set to "text/javascript".

**Placement of JavaScript in HTML file**

There is a flexibility given to include JavaScript code anywhere in an HTML document. The most preferred ways to include JavaScript in your HTML file are:

* <HEAD>...</HEAD> section.
* <BODY>...</BODY> section.
* In an external file.

**1.<HEAD>...</HEAD> section**

Syntax:

<HTML>

<HEAD>

<SCRIPTtype="text/javascript">

</SCRIPT>

</HEAD>

<BODY></BODY>

</HTML>

**Example: 1**

<HTML>

<HEAD>

<SCRIPT

TYPE="text/javascript">

function Welcome(){

alert("Welcome !")}</SCRIPT>

</HEAD>

<BODY>

<CENTER><INPUT

TYPE="button"onclick="Welcome

()" value="Click

Me"></CENTER>

</BODY>

</HTML>

**2. <BODY>...</BODY> section**

Syntax:

<HTML>

<HEAD></HEAD>

<BODY>

<SCRIPT type="text/javascript“>

<P>This is a paragraph</P>

</BODY>

</HTML>

**Example: 2**

<HTML>

<BODY>

<SCRIPT

TYPE="text/javascript">

function Welcome() {

alert("Welcome !") }

</SCRIPT>

<CENTER>

<INPUT TYPE="button"

onclick="Welcome()"

value="Click Me">

</CENTER>

</BODY>

</HTML>

**3. In an external file**

Syntax:

<HTML>

<HEAD>

<SCRIPT type="text/javascript"

src="filename.js”></SCRIPT>

</HEAD>

<BODY></BODY>

</HTML>

**ORACLE 21C:**

**Introduction of oracle 21c database**:

Oracle 21c database is the cloud version of oracle. Oracle Database 21c, the latest version of the world’s leading converged database which is available on Oracle Cloud, including the Always Free tier of Oracle Autonomous Database. Oracle Database 21c contains more than 200 new innovations, including immutable blockchain tables, In-Database JavaScript, native JSON binary data type, AutoML for in-database machine learning (ML), and persistent memory store, as well as enhancements for in-memory, graph processing performance, sharing, multitenant, and security. Unlike other vendors’ single-purpose databases in the cloud or on-premises, Oracle Database 21c provides support for multi-model, multi-workload, and multi-tenant requirements – all within a single, modern converged database engine. In addition, Oracle today announced the availability of Oracle APEX (Application Express) Application Development, a new low-code service for developing and deploying data-driven enterprise applications quickly and easily. The browser-based, low-code cloud service enables developers to create modern web apps for desktops and mobile devices using an intuitive graphical interface.

Oracle Database 21c is the database engine that powers Oracle database services in the cloud and on-premises, including Oracle Autonomous Database, Oracle Exadata Database Service, Oracle Exadata Database Cloud Customer, and Oracle Exadata Database Machine. The latest release includes more than 200 new innovations, which extend database convergence to new use cases, optimize performance, and improve developer, analyst, and data scientist productivity. Key innovations include:

**Immutable Blockchain Tables:** Blockchain Tables bring the key security benefits of blockchain technology to enterprise applications. Part of Oracle’s Crypto-Secure Data Management, Blockchain Tables provide immutable insert-only tables whose rows are cryptographically chained together. By providing tamper detection and prevention capabilities directly in the Oracle Database, customers can protect against illicit changes by insiders or hackers impersonating administrators or users. Blockchain Tables are part of the converged database, accessed with standard SQL, and support full analytics and transactions—making it orders of magnitude easier to use, and more functional, than existing blockchain implementations. Blockchain Tables are a free feature in all Oracle Database editions.

**Native JSON Data Type:** Oracle has provided powerful SQL/JSON query and indexing support for many years. Database 21c adds a new JSON data type representation, enabling up to 10x faster scans and up to 4x faster update operations. Overall, these improvements make Oracle SQL/JSON 2x faster than MongoDB and AWS Document DB on the YCSB benchmark. As with previous releases, users can mix or join JSON and other data types; index any JSON element for fast OLTP; use declarative parallel SQL analytics across all formats; and run complex joins across multiple JSON documents and collections—all without any need for custom application code.

**AutoML for In-Database Machine Learning:** Automatically builds and compares machine-learning models at scale, and facilitates the use of machine learning by non-experts. A new AutoML user interface makes it easier for non-expert users to leverage in-database machine learning. Oracle also added new algorithms for anomaly detection, regression, and deep learning analysis to our extensive library of popular, in-database machine learning algorithms.

**In-Database JavaScript:** Enables developers to work efficiently in modern programming languages. The embedded Grail Multilingual Engine allows JavaScript data processing code to run inside the database – where the data resides – eliminating expensive network round-trips. In addition, users can easily execute SQL from within JavaScript code, and JavaScript data types are automatically mapped to Oracle Database data.

**Persistent Memory Support:** Stores database data and redo logs in local Persistent Memory (PMEM), which significantly improves the performance of IO-bound workloads. SQL runs directly on data stored in the direct-mapped Persistent Memory file system, eliminating the IO code path and the need for large buffer cache. In addition, new database algorithms prevent partial or inconsistent stores to Persistent Memory.

**Higher Performance Graph Models:** Allows modelling of data based on relationships, and enables exploration of connections and patterns in social networks, IoT, and more. Further improvements in memory optimization reduce the amount of memory required to analyze larger graphs, which enables existing applications to run faster with no changes. In addition, users can create or extend graph algorithms using Java syntax, which can execute as native algorithms since they are compiled with the same optimizations.

**Database In-Memory Automation:** Oracle supports both row and column formats in the same table to allow analytics and transactions to run simultaneously on the same table. Oracle Database 21c introduces a Self-Managing In-Memory Column Store that simplifies and improves efficiency by automatically managing the placement and removal of objects in the In-Memory Column Store, then tracks usage patterns and moves and evicts objects from the column store. In addition, columns are automatically compressed based on usage patterns. Oracle Database 21c also introduces new in-memory vector join algorithms to speed up complex queries.

**Shading Automation:** Native Database Shading delivers hyperscale performance and availability while enabling global enterprises to easily meet data sovereignty and data privacy regulations. Data shards share no hardware or software, and can reside on-premises or in the cloud. To simplify the design and use of sharing, Oracle Database 21c includes a Shading Advisor Tool that assesses a database schema plus its workload characteristics and then provides a sharded database design optimized for performance, scalability, and availability. Backup and Recovery across shards is also automated.

**Role of Oracle 21c Database in Oracle Apex:**

Oracle 21c is the latest version of Database by Oracle which contains very broad field of features in itself. Apex uses Oracle 21c as Back-End to Interact with Databases through SQL commands. Oracle 21c uses 2 language generally for Insert Update Delete Create operations with some Validation and Actions these languages are SQL and PL/SQL.

Oracle Apex Featured Graphical User Interface to create Tables and triggers and all the Things which are needed for create database for Any webApp with Cloud Storage as Well as Command Line Interface where we can write SQL commands to perform any operation.

GUI is basically The Interface which is Easier to perform any operation without using Hundreds of lines of codes. It may also easier to Non-Technical person Where CLI is used to perform some extra and rare commands which are not featured by GUI for Example creating a sequence.

For both GUI and CLI, Developer should have Basic Knowledge of SQL and Pl/SQL

And the basic knowledge of Database for using the features provided by Oracle 21c.

**SQL:**

Structured Query Language or SQL is a standard Database language which is used to create, maintain and retrieve the data from relational databases like MySQL, Oracle, SQL Server, Postgrad, etc.

• MySQL is an open-source relational database management system (RDBMS). Its name is a combination of “My”, the name of co-founder Michael Wideness’ daughter, and “SQL”, the abbreviation for Structured Query Language.

• It is written in C & C++

**Features:**

• Cross –platform support

• Triggers

• ACID (Atomicity, Consistency, Isolation, Durability) rule

• GUI Support

• MySQL workbench is the integrated environment for MySQL. It enables users to graphically administer MySQL databases and visually design database structures.

**PL/SQL:**

PL/SQL is a procedural Language designed specially to embrace SQL statements with in its Syntax. PL/SQL program units are compiled by the Oracle Database server and stored inside the database. At the run time, both SQL and PL/SQL run within the same server process, bringing optimal efficiency. PL/SQL automatically inherits the robustness, security and portability of the Oracle Database.

PL/SQL is the Block Structured language means that the PL/SQL programs are divided and written in logical blocks of code. Each Block Consist of three sub Parts:-

**1. Declaration:**

This section starts with the keyword DECLARE. It is an Optional section and Defined all Variables, cursors, subprograms, and other element to be used in the program.

**2.Executable Commands:**

This section is enclosed between the keywords BEGIN and END and it is a mandatory section. It consist of the PL/SQL statements of the program. It should have at least one executable line of code, which may be just a null command to indicate that nothing should be executed.

**3.Exception Handling:**

This section starts with the keyword EXCEPTION. This optional section contains exception(s) that handle errors in the program.

Every PL/SQL Statement ends with semicolon (;) PL/SQL blocks can be nested within other PL/SQL blocks using BEGIN and END. Following is the Basic Structure of PL/SQL blocks-

DECLARE

<declaration section>

BEGIN

<executable command(s)>

EXCEPTION

<exception handling>

END;

**Example:-**

DECLARE

Message varchar2(20): = ‘HELLO WORLD’;

BEGIN

Dbms\_output.put\_line(message);

END;

/

**CHAPTER 2**

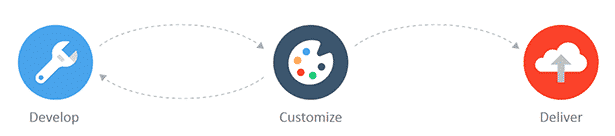
**VALUE OF ORACLE APEX APPLICATION DEVELOPMENT**

**OVERVIEW:**

Oracle APEX (also known as APEX or Oracle Application Express) is an enterprise low-code development platform from Oracle Corporation that is used to develop and deploy web applications on Oracle databases. APEX provides a web-based integrated development environment (IDE) that uses wizards, drag-and-drop layout, and property editors to build applications and pages.

APEX intends to simplify the process of creating web applications that serve as a frontend to a database server, among other things.

APEX is a fully-supported no-cost feature of the Oracle Database and can be installed anywhere Oracle Database runs. APEX is also offered on Oracle's Cloud across various services including Autonomous Database Cloud Services and the stand-alone APEX Application Development service.



**RELEASES/HISTORY:**

Oracle Apex is developed from HTML DB in year 2006.

**BACKGROUND:**

Oracle APEX has gone through many name changes since its inception in 2000. Names include:

**Flaws:**

* Oracle Platform
* Project Marvel
* HTML DB
* Application Express (APEX) aka Oracle APEX

APEX was created by [Mackey McCandless](https://www.google.com/search?rlz=1C1YTUH_enIN1015IN1015&q=Mackey+McCandlish&stick=H4sIAAAAAAAAAONgVuLSz9U3iK8oiDcqe8Royi3w8sc9YSmdSWtOXmNU4-IKzsgvd80rySypFJLgYoOy-KR4uJC08SxiFfRNTM5OrVTwTXZOzEvJySzOAACSBt0KWgAAAA&sa=X&ved=2ahUKEwiQ8evMxMj6AhXw-3MBHcLTDMgQzIcDKAB6BAgXEAE), a developer at Oracle, after development of his previous project, Web DB, started to diverge from his original vision. Although APEX shares some functionality with Web DB, it was developed from scratch and there's no upgrade path from Web DB to APEX. When tasked with building an internal web calendar, Hichwa enlisted the help of Joel Kallman and started development on a project called Flows. Hichwa and Kallman co-developed the Web Calendar and Flows, adding features to Flows as they needed them to develop the calendar. Early builds of Flow had no front-end so all changes to an application had to be made in SQL\*Plus via inserts, updates and deletes.

APEX is extensively used internally by Oracle to develop its support sites. The AskTom knowledgebase, online store, Dev Gym, and livesql run on APEX.

**WHY WE CHOOSE ORACLE APEX ?**

Because it will enable you to get more done in less time.

The digital economy is not kind to those who move slowly. With technology-based business models becoming the norm across a broad range of industries, whether a company “lives or dies” is increasingly becoming dependent on shortening time-to-market and responding quickly to shifting customer expectations. Organizations need to both adapt quickly and iterate quickly to keep up, and this is exactly where APEX shines.

As a growing number of organizations discover how simple and easy it is to build highly functional web and mobile applications using this deceptively powerful tool, it’s no wonder that the popularity of APEX has shot up in the last few years.

**WHY ORACLE APEX IS MUCH IN DEMAND?**

**It’s free to use:** If you have an Oracle database, you already own APEX. Using it won’t cost you anything.

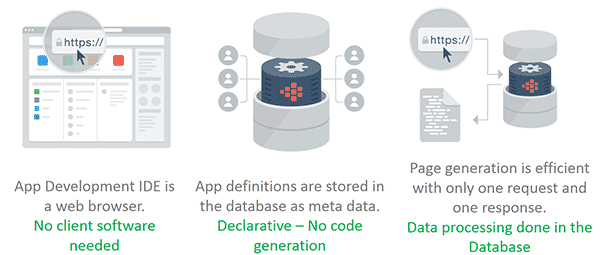
**It’s easy to get started:** There are no complex programs to install on your computer before you get started. All you need is a web browser.

**You don’t need to be a “coder”:** With its easy-to-use interface where you can “point and click” and “drag and drop”, you can build amazing applications without writing any code. In fact, Oracle APEX comes with a host of productivity applications you can install and configure to your own needs. They're a great way to learn, as well!

**You can focus on your problem:** With the hairy details of web applications such as security, user management, access control, pagination and navigation all taken care of for you behind the scenes, you can focus on building your requirements and not get bogged down with implementation details.

**Your applications will look great:** APEX comes with a highly-configurable universal theme, which includes responsive design templates, grid-layout for forms, and the ability to configure or choose from a number of pre-defined style and display options. So, without being a designer, you can simply use the defaults and build really professional looking applications with rich functionality that are intuitive for users and have a modern look and feel. And, best of all, with the built in responsive design, your apps will look good on any device.

**There’s a great community to help you:** With over 400,000 developers, 75+ bloggers, and very active community forums, you won’t have any trouble finding help when you get stuck. Be sure to check out apex. World, and also look for meetups in your area.



**WHY ENTERPRISES WANT ORACLE APEX?**

**APEX drastically improves delivery times:** With its simple browser based development environment, self-service provisioning, and many built-in components for building rich forms and reports, applications can be developed and deployed in record time.

**APEX is robust, scalable, and secure:** Since APEX is built right into the Oracle database and designed with exceptional efficiency, it is as scalable as the database itself. Failover and redundancy is automatically taken care of as part of standard Oracle Database functionality, and resource consumption is governed by native Oracle Database features (Database Resource Manager). And, you can leverage the built-in Oracle database security features and options, such as Database Vault, Advance Security Option, and Redaction, and also use the native “Advisor” to highlight any security issues.

**APEX enforces consistency and reduces chance of error:** It's a framework with many configuration options to control how applications are developed, as well as the pre-integrated standard libraries and user interface components, it’s easy to ensure applications have a consistent look and feel, which significantly improves user adoption. And, since it’s a framework with a ton of built in functionality, there is less code to write, which means less chance of error and greater productivity.

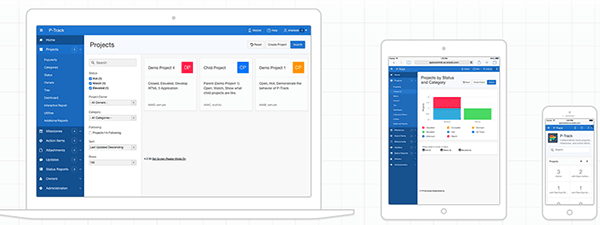
**You can focus on your problem:** With the hairy details of web applications such as security, user management, access control, pagination and navigation all taken care of for you behind the scenes, you can focus on building your requirements and not get bogged down with implementation details.

**APEX is a team “force multiplier**”: APEX opens up application development to a larger group of contributors. Typical web and mobile development requires a unique set of skills that is not always easy to find. With APEX, traditional Oracle developers can leverage their SQL and PL/SQL skills to become immediately productive web and mobile application developers.

**Your applications will look great:** APEX comes with a highly-configurable universal theme, which includes responsive design templates, grid-layout for forms, and the ability to configure or choose from a number of pre-defined style and display options. So, without being a designer, you can simply use the defaults and build really professional looking applications with rich functionality that are intuitive for users and have a modern look and feel. And, best of all, with the built in responsive design, your apps will look good on any device.

**APEX is Aligned With Industry trends:** APEX fosters agile development practices and has strong support for modern and responsive application technologies, including HTML5, CSS3, integrated javascript libraries and Restful web services and perhaps most importantly, it’s cloud ready.

**Productivity Applications right out of the box!** Oracle APEX comes with professional grade productivity apps you can start using in your business now. Install them, use them 'as-is', or customize to your own needs to leverage maximum efficiency. They're also a great way to learn Oracle APEX best practices, and get started quickly with this powerful low code platform

****

**DEMAND OF ORACLE APEX IN FUTURE:**

Today, two modes or two speeds of IT Development exist within enterprise companies; this modern-day development is defined by Gartner as bimodal IT. The first mode is the inexpensive, fast paced development is completed by tech giants like Google and startups like, Slack. Developers in this fast paced mode are constantly releasing small improvements to the system which allows for the ability to quickly respond to user’s needs.

On the contrary, there’s the massive, costly, and traditional enterprise development. This process is slow-moving for good reasons. When dealing with large companies, such as banks, a small error can be costly. Airtight and careful development is an absolutely must. Achieving business objectives quickly can be hindered by this process.

Oracle is another example of a company that can be grouped in the slower speed. There’s an unfortunate stigma attached to the company – Oracle’s development is timely and costly. Nonetheless, the end results are very large products with stability and security. However, within the Oracle database is a fast, web-based development tool called Application Express (Apex). APEX is no-cost option to the database; it’s already included. Oracle APEX truly facilitates rapid application development.

**DEMAND OF ORACLE APEX IN ANY ENTERPRISE (DEVELOPMENT SOLUTION):**

Oracle APEX provides solutions for both modes of development and all types of businesses. As APEX already resides within the database, the business has the liberty to determine which development mode to apply on their data rather than which tool. This is a major advantage when compared to other development platforms which may only work well with one of the two modes and may require a team to know two different languages.

Martin speaks about APEX development after building many real world applications. His development experience varies from small to medium businesses, major US universities, and large international companies in oil and gas or pharmaceuticals. The development is rapid enough for the smaller companies, yet secure and functional enough for the large enterprises.

**RAPID DEVELOPMENT WITH AMAZING FUNCTIONALITY:**

Oracle APEX gives the end-user powerful functionality provided by super quick development. Furthering the feature-rich abilities of APEX are the plugins options. An online community, APEX. World, is a space where developers worldwide are creating and sharing innovations to this same application. The site hosts diverse plug-ins all variated from application. Martin demonstrates the ease of installing a plug-in.

The plug-ins provide amazing functionally, which would take days or weeks to develop, but there are smart APEX developers out there, making life much easier for the APEX community. This functionality is incredibly useful for the end-user, and requires minimal effort for developers to implement, thus, reducing the cost to the business.

There’s a lot of cool things you can do with APEX. It makes the developer look good with minimal efforts and saves the business money by reducing overall development costs. APEX works for both modes of IT – valuable for slower moving companies which require the consistency and security, but ideal for more agile organizations looking to quickly display front-end results for users. Leveraging the power APEX can truly boost productivity and create dynamic functionality. If your business is rapidly moving, rapidly development, Oracle APEX can still work within those bounds and do a lot more.

**CHAPTER 3**

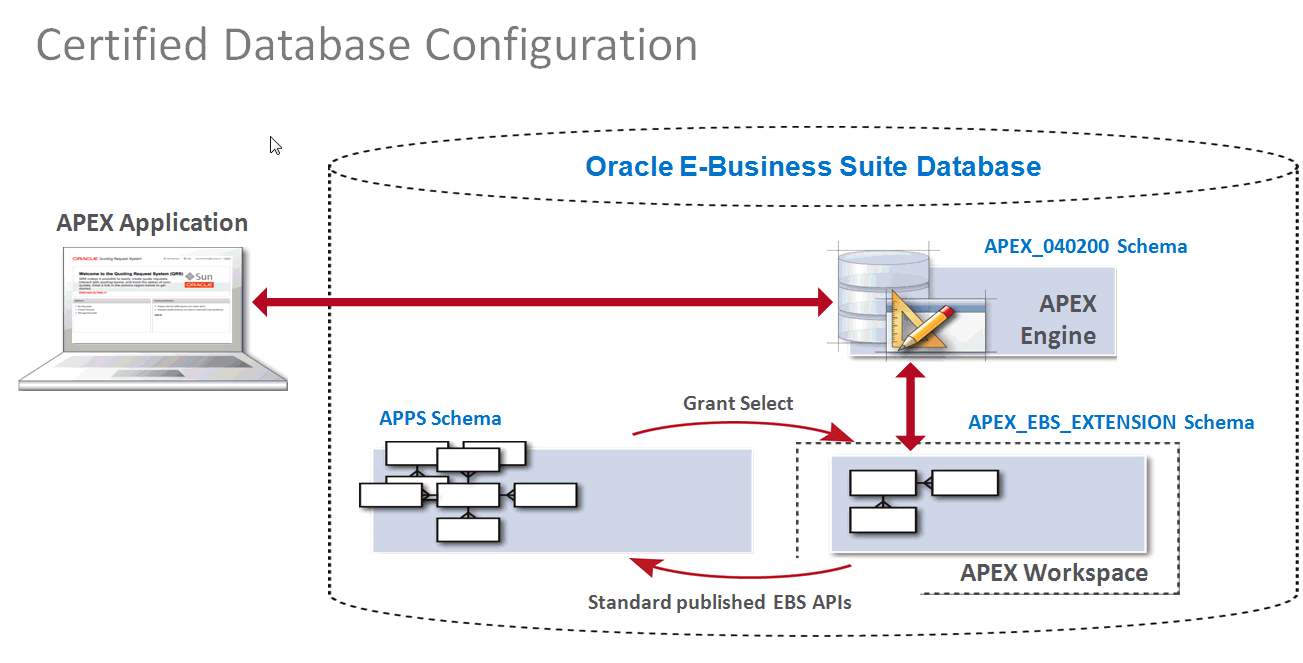
**ANALYSIS OF ORACLE APEX**

Oracle APEX installs with your Oracle database and consists of data in tables and PL/SQL code. Whether you run the APEX development environment or run an application built using APEX, the process is the same. Your browser sends a URL request that is translated into the appropriate APEX PL/SQL call. After the database processes the PL/SQL, the results are relayed back to your browser as HTML. This cycle happens each time you either request or submit a page.

APEX does not use a dedicated database connection. Instead, each request is made through a new database session, consuming minimal CPU resources. Application session state is managed in the database tables by the APEX engine.

Behind the scenes, the APEX engine renders and processes pages. The APEX engine also performs these tasks:

* Session state management
* Authentication services
* Authorization services
* Page flow control
* Validation processing



Oracle APEX uses a simple 3-tier architecture where requests are sent from the browser, through a web server, to the database. All processing, data manipulation and business logic is executed in the database. This architecture guarantees zero latency data access, top performance, and scalability, out of the box.

A web request from the web browser is sent to Oracle REST Data Services (ORDS) where it is handed to Oracle Database to be actioned. Within the database, the request is processed by Oracle APEX. Once the processing is complete, the result is sent back through ORDS to the browser.

**THE ORACLE RAD STACK:**

The Oracle RAD stack is an inclusive technology stack based on three core components: Oracle REST Data Services (ORDS), Oracle APEX, and Oracle Database.

This stack provides all the necessary components to develop and deploy world-class, powerful, beautiful, and scalable apps. There are no other moving parts of additional components required. In addition, both Oracle APEX and ORDS are no-cost features of Oracle Database, meaning if you have Oracle Database, you already have this Oracle RAD stack.

**REST Data Services (ORDS)**

ORDS is a Java application that enables developers with SQL and database skills to develop REST APIs for Oracle Database, Oracle Database 12c JSON Document store and higher, and the Oracle NoSQL Database.

**APEX**

The Oracle Database's native low-code development platform that enables you to build stunning, scalable, secure apps, with world-class features, that can be deployed anywhere.

**Database**

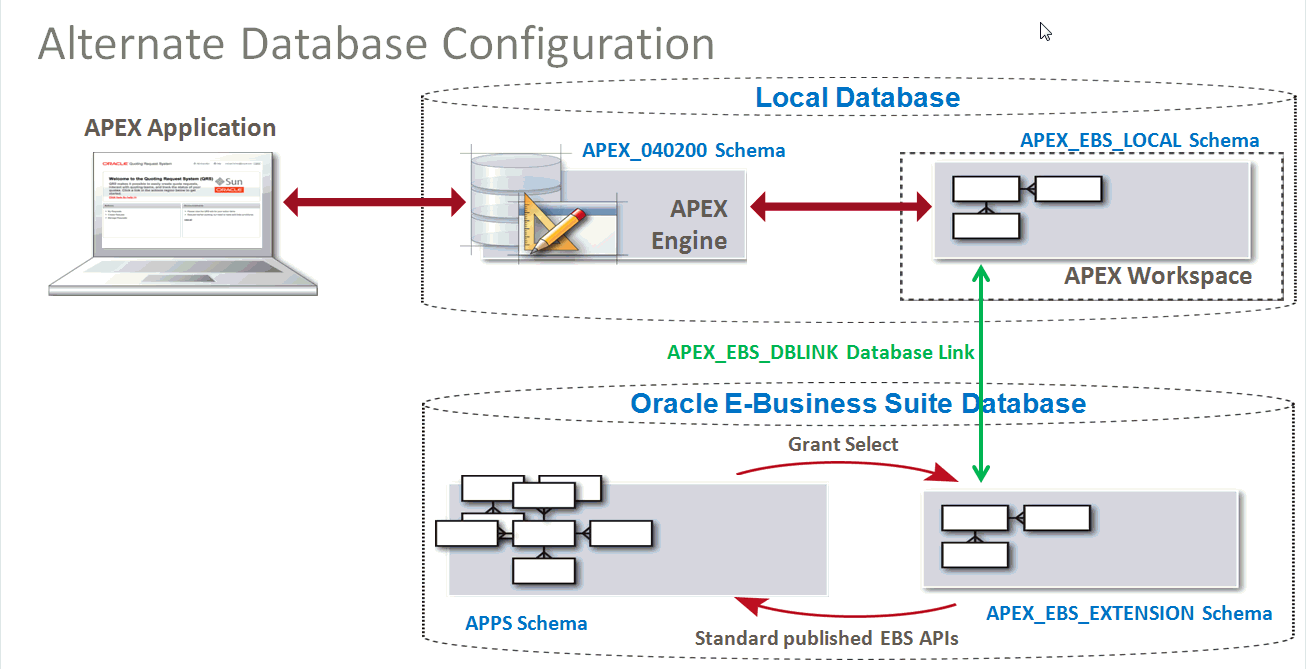
Oracle Database — the most complete, integrated, and secure database solution for any scale deployment. This solid foundation enables apps built using Oracle APEX to be enterprise ready from day one.

**MATADATA DRIVEN:**

When you create or extend an application, Oracle APEX creates or modifies the metadata stored in its database tables. When the application is run, the Oracle APEX engine then reads the metadata and displays the requested page or processes page submissions.

To provide stateful behavior within an application, Oracle APEX transparently manages session state in the database. Application developers can get and set session state using simple substitutions as well as standard SQL bind variable syntax. There is no need for file-based compilation and there is no code generation.

All processing is performed by PL/SQL acting directly on the data schemas in the database. Therefore, Oracle APEX apps are very efficient as data is manipulated directly in the database, and the results are sent back to your web browser. A single API call invokes all of the necessary data processing, based on the metadata definition, in a single request rather than requiring multiple calls to the database.



**CHAPTER 4**

**PROJECT DEMONSTRATION AND IMPLEMENTATION**

“Library Management System” is the Project which is developed through Oracle Apex at Front-End in which Oracle Apex tools and HTML, CSS are Used & at Back-End oracle 21c is Used. Oracle 21c is a cloud version of Database in which SQL and pl/SQL Languages are used.

“Inventory Management System” is the project that Contains Much of the services provided new member add new user and customer, add admin, add customer,manage inventory , mange Branch, manage Dept etc.

**SOME SNAPSHOTS OF INVENTORY MANAGEMENT SYSTEM:**

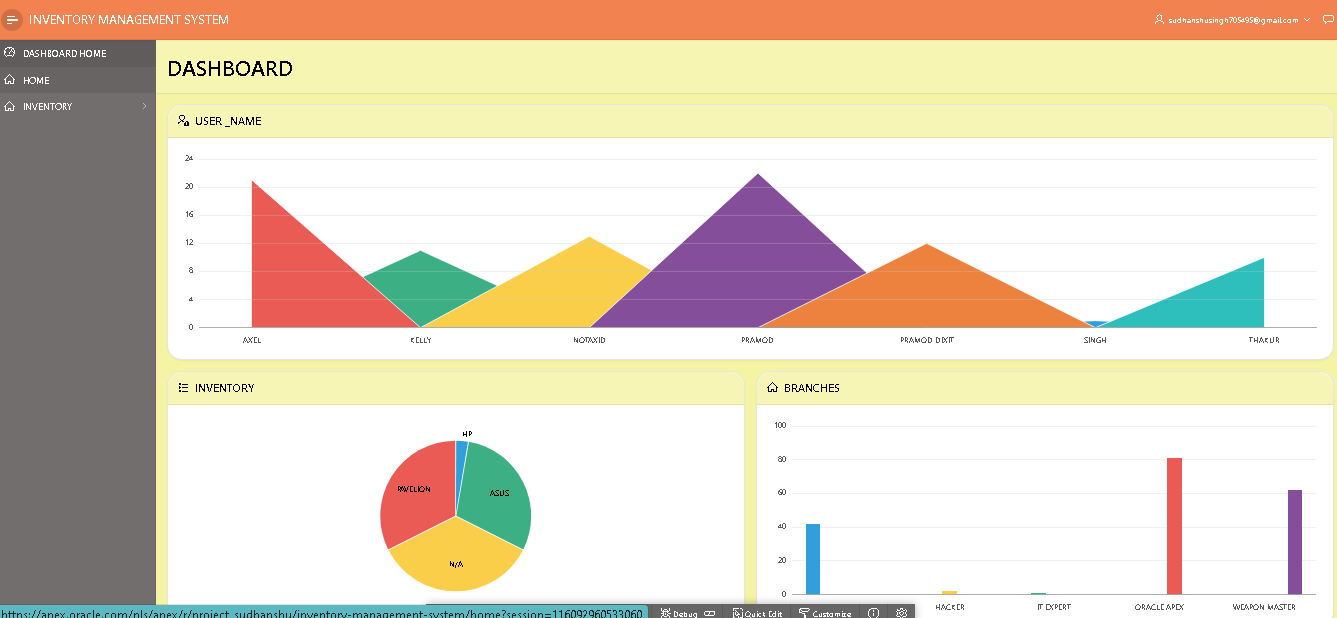
**Here some snapshot of Inventory Management System.**

****

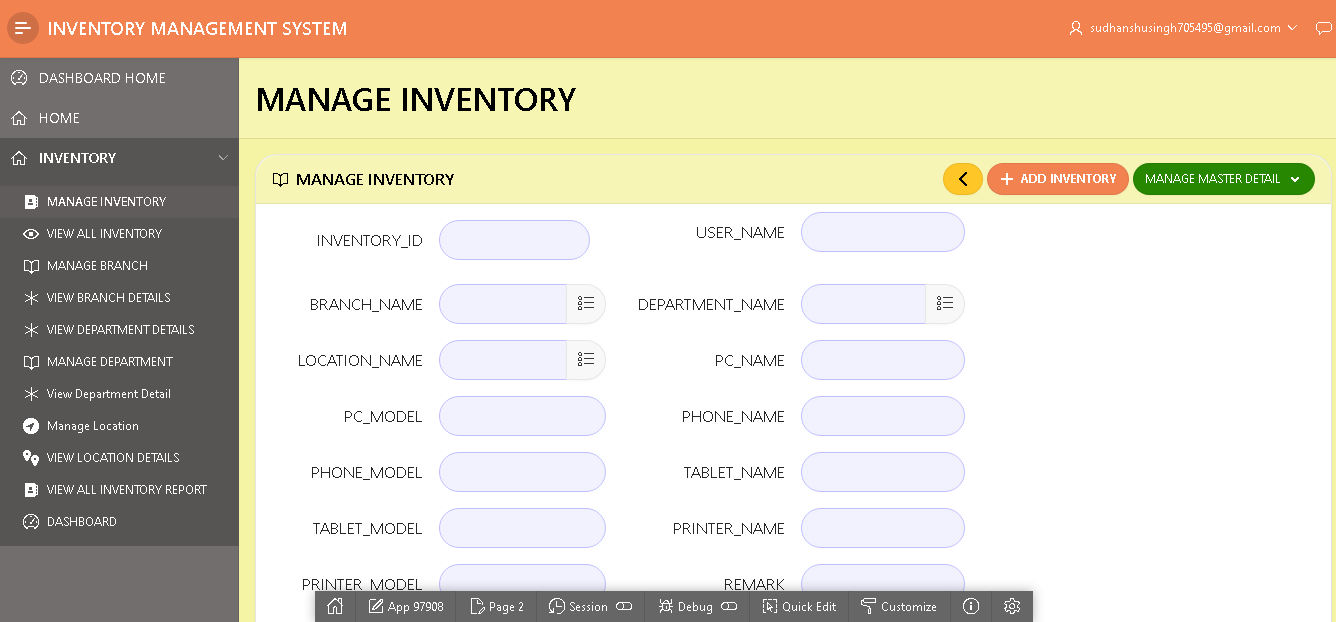
**Home Page Of Inventory Management System**

****

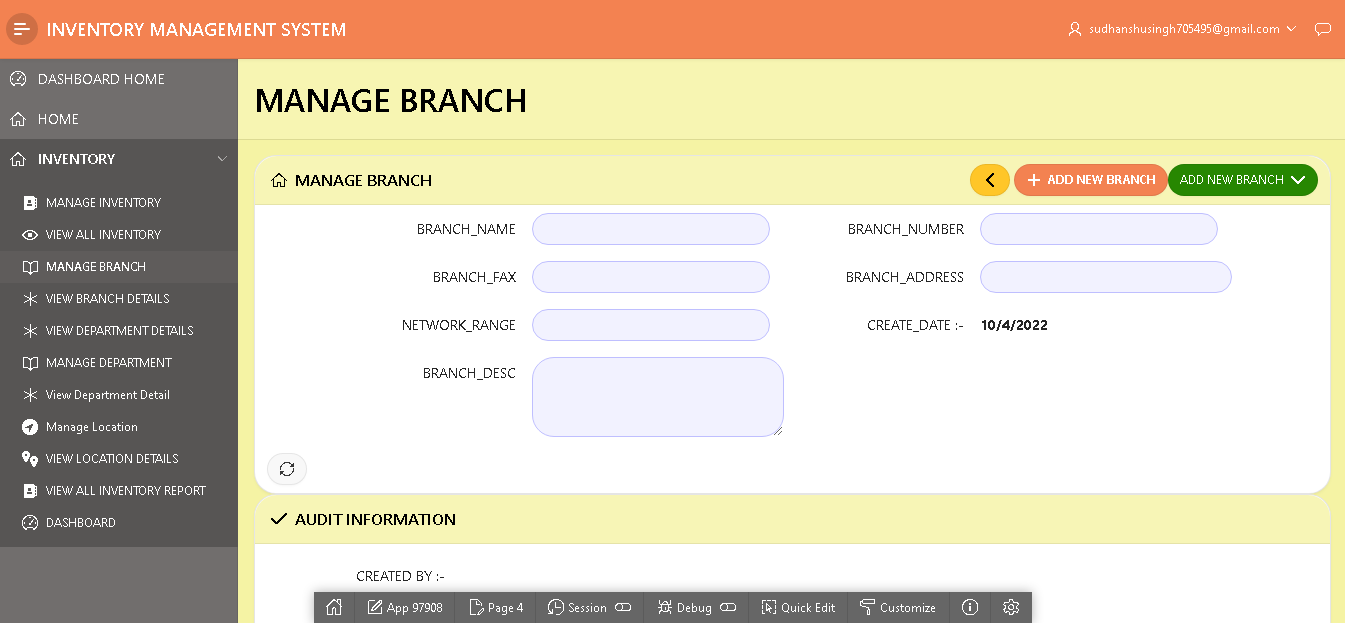
**Add Dash Board**

****

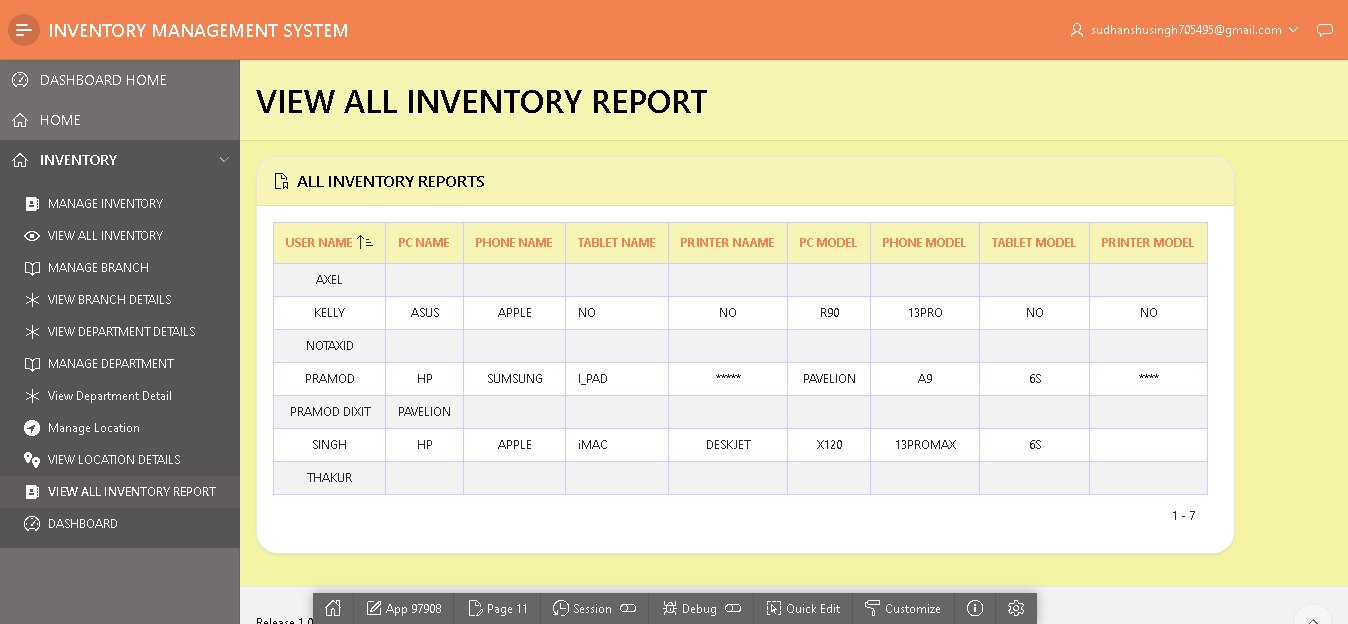
**Inventory System**

****

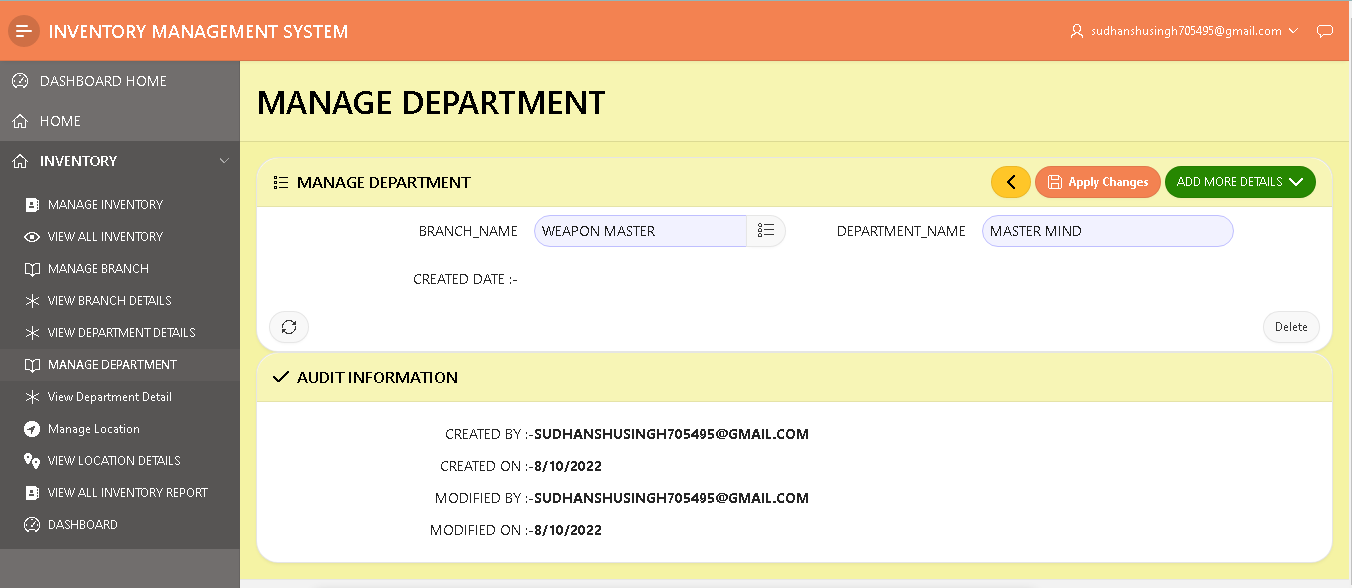
**Inventory Branch**

****

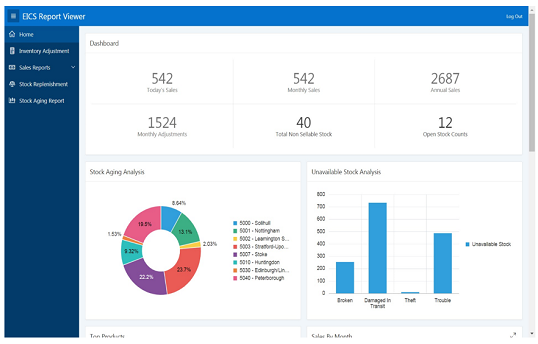
**Admin Report**

****

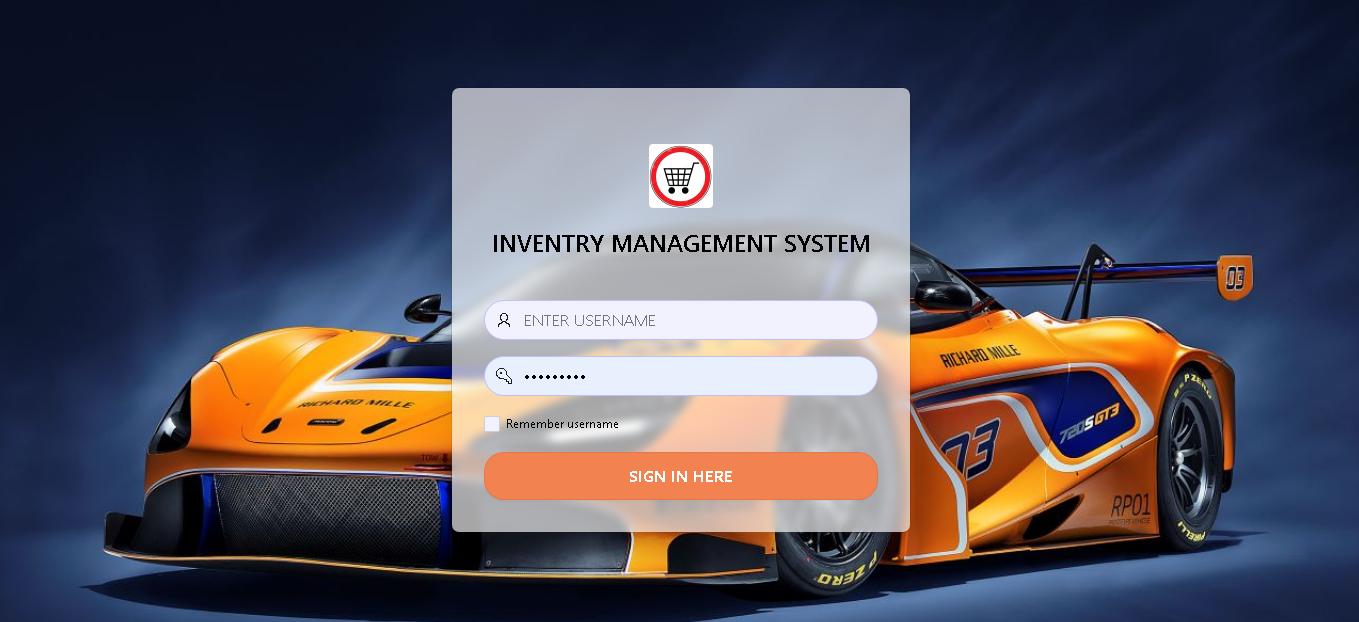
**View Inventory Details**

****

**Report View**

****

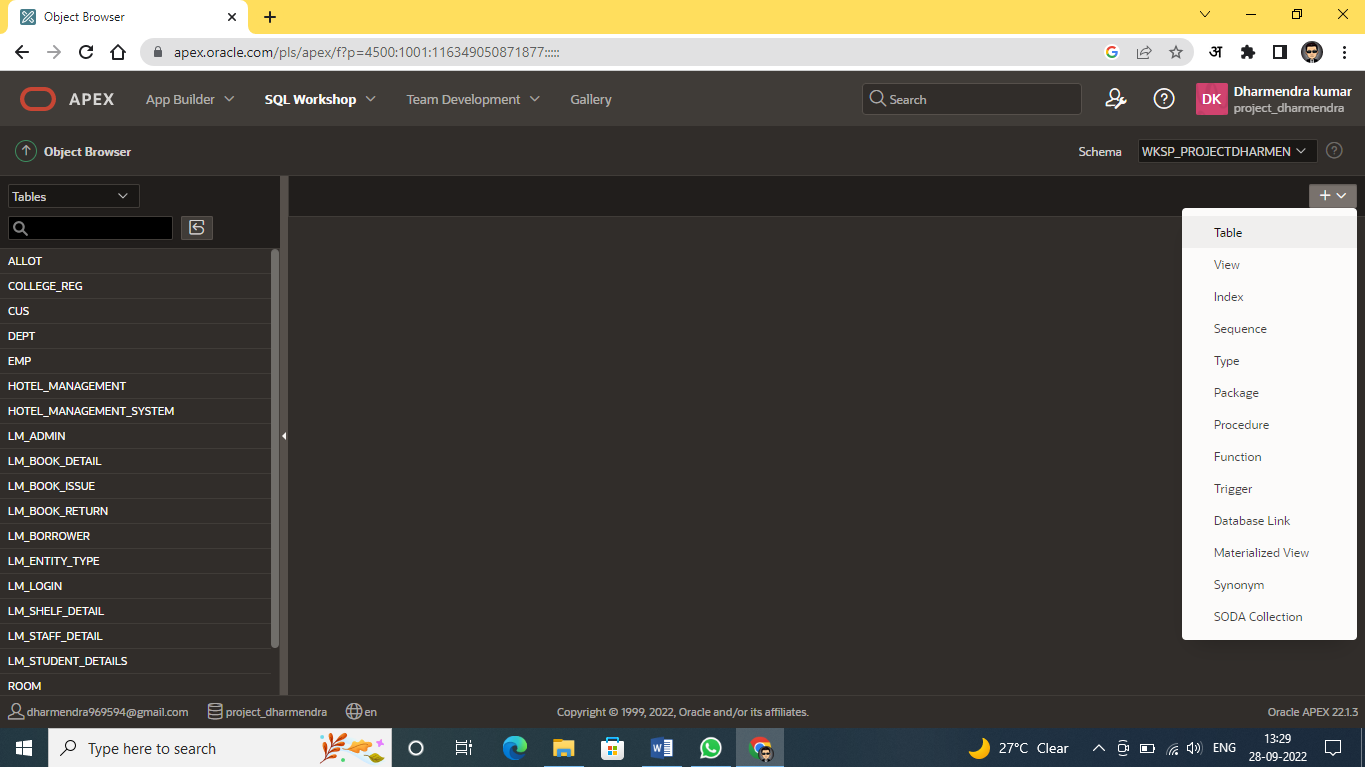
**Login Page**

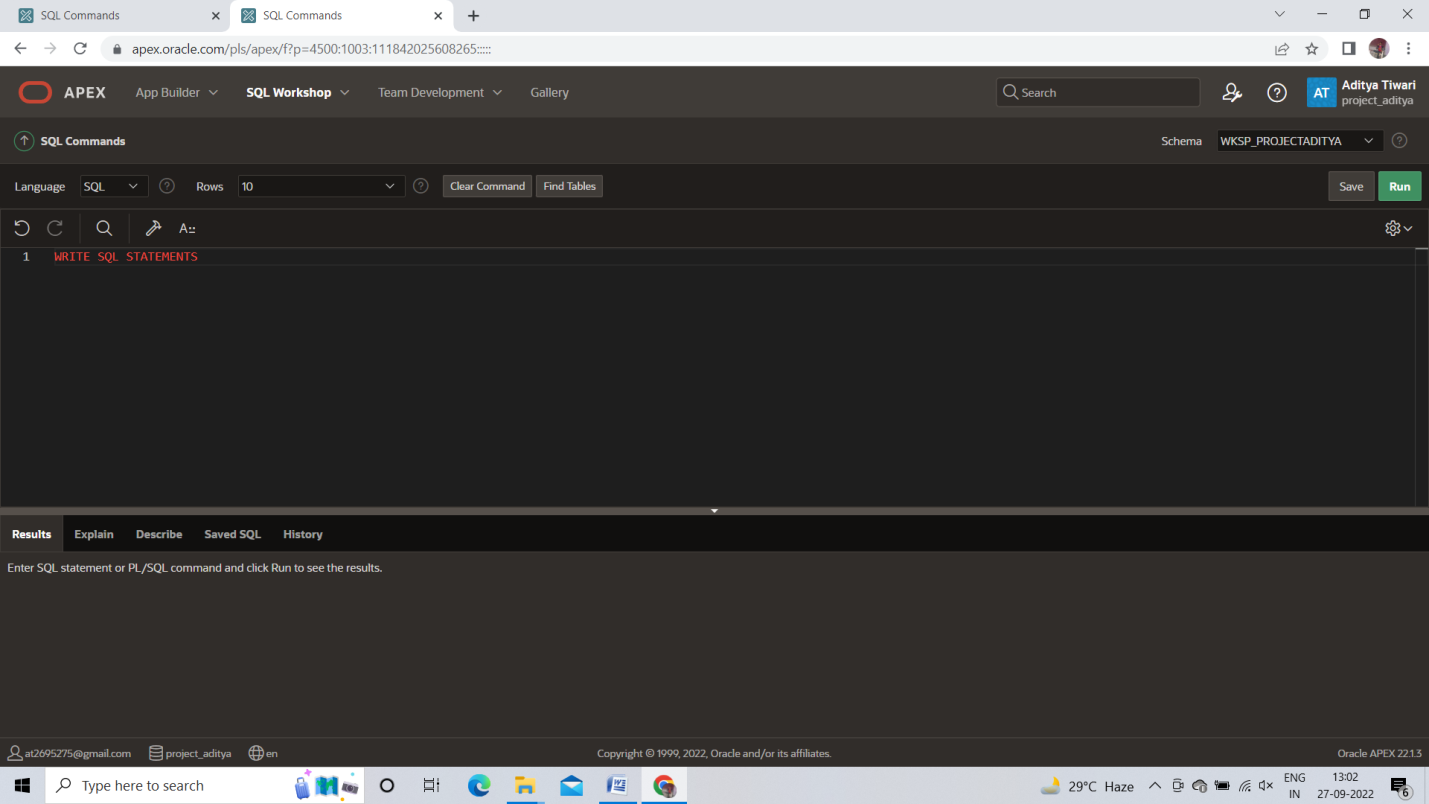
****

**IMPLEMENTATION**

OPEN BROWSER SEARCH ORACLE APEX AND LOGIN WITH IT IF YOU HAVE ALREADY ACCOUNT IF YOU DON’T HAVE ANY ACCOUNT REGISTER AND CREATE ACCOUNT

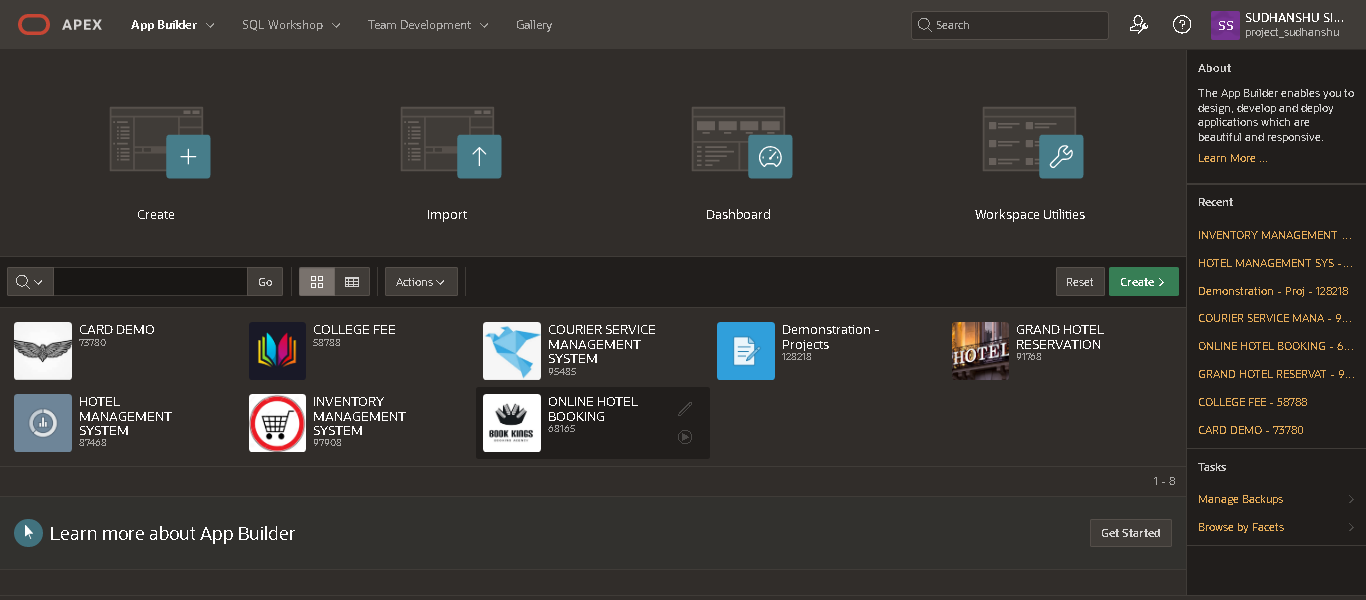
**CREATE DATABASE BY SQL COMMANDS OR BY USING OBJECT BROWSER:-**



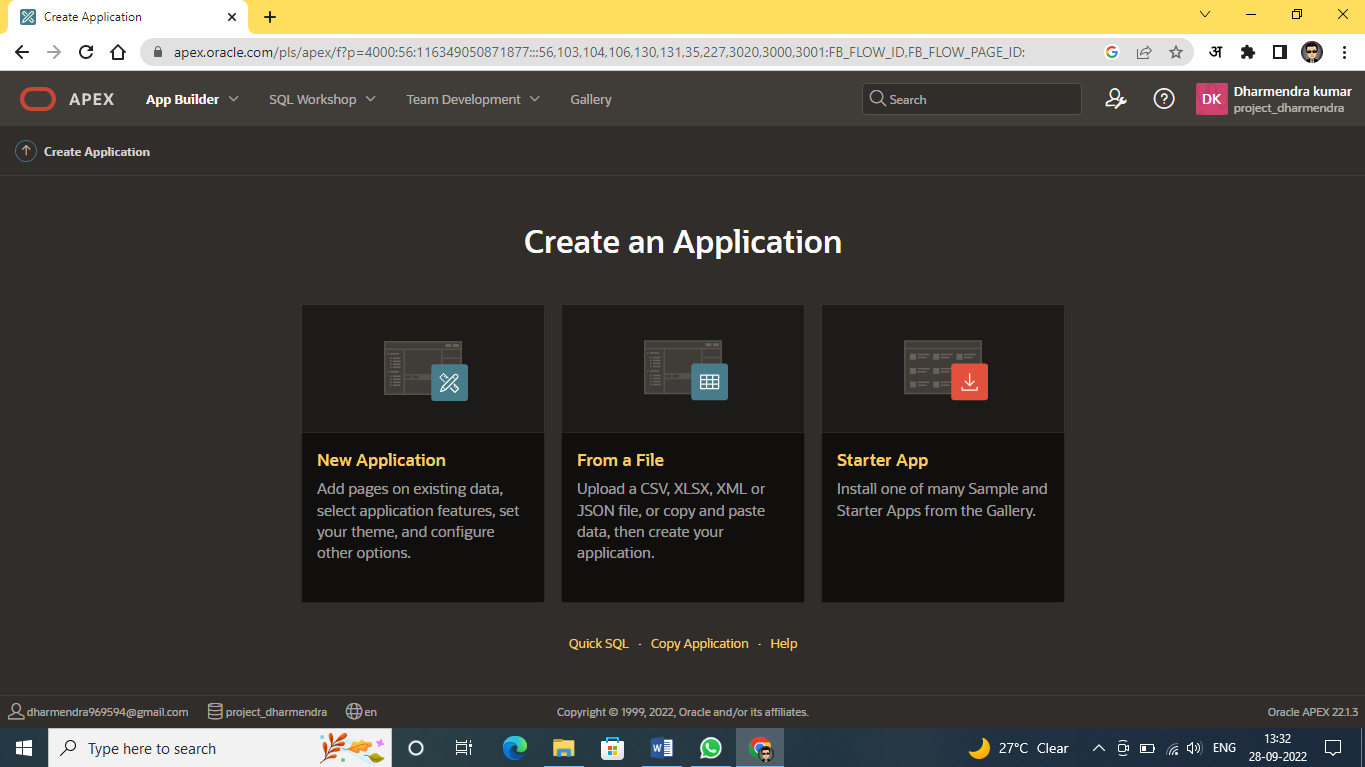


**CREATE FRONT-END BY APP BUIDER:-**

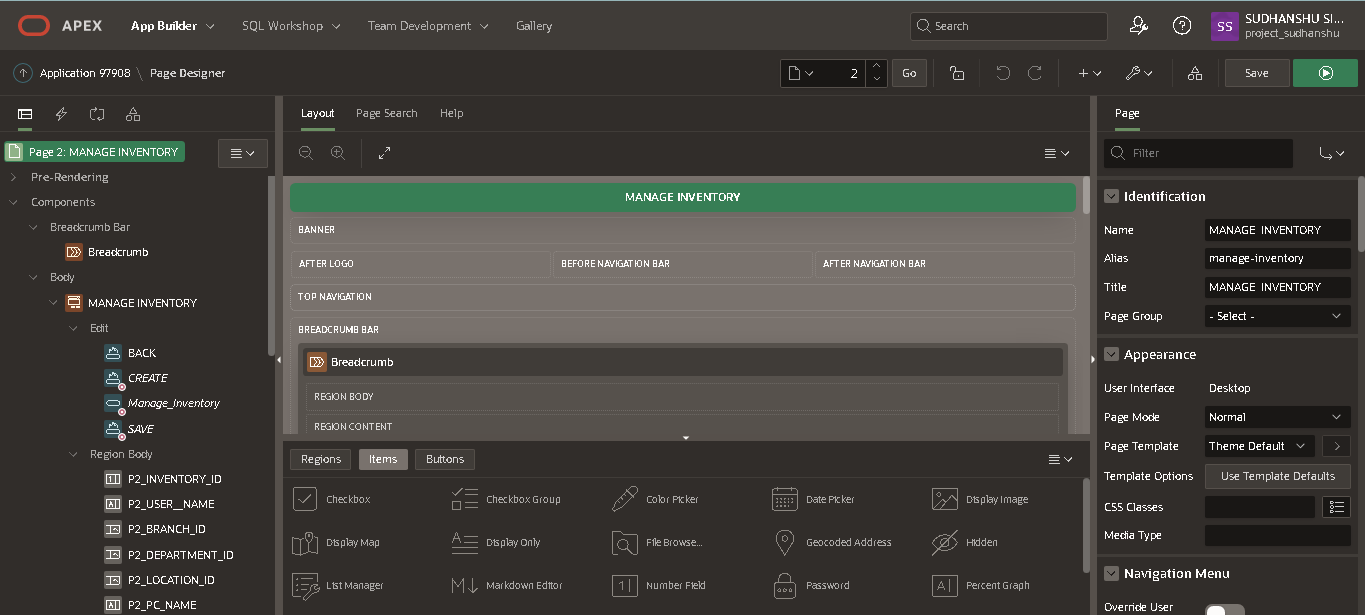
**Go to app builder click on create**



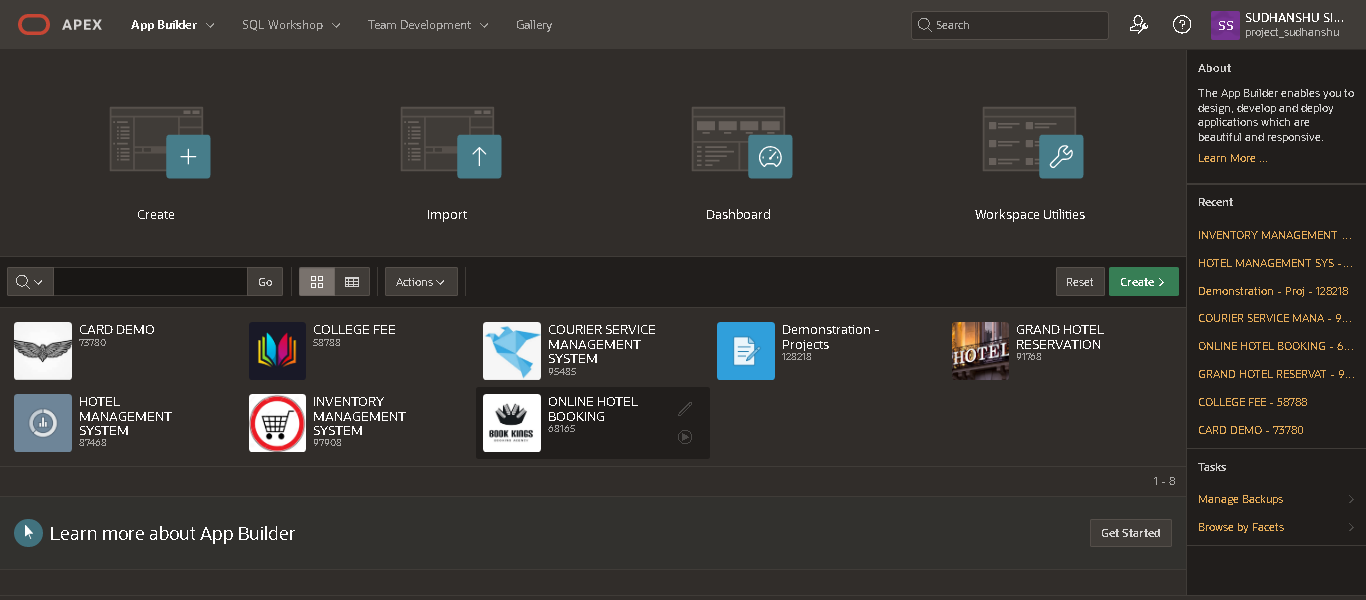
**Click on New Application**



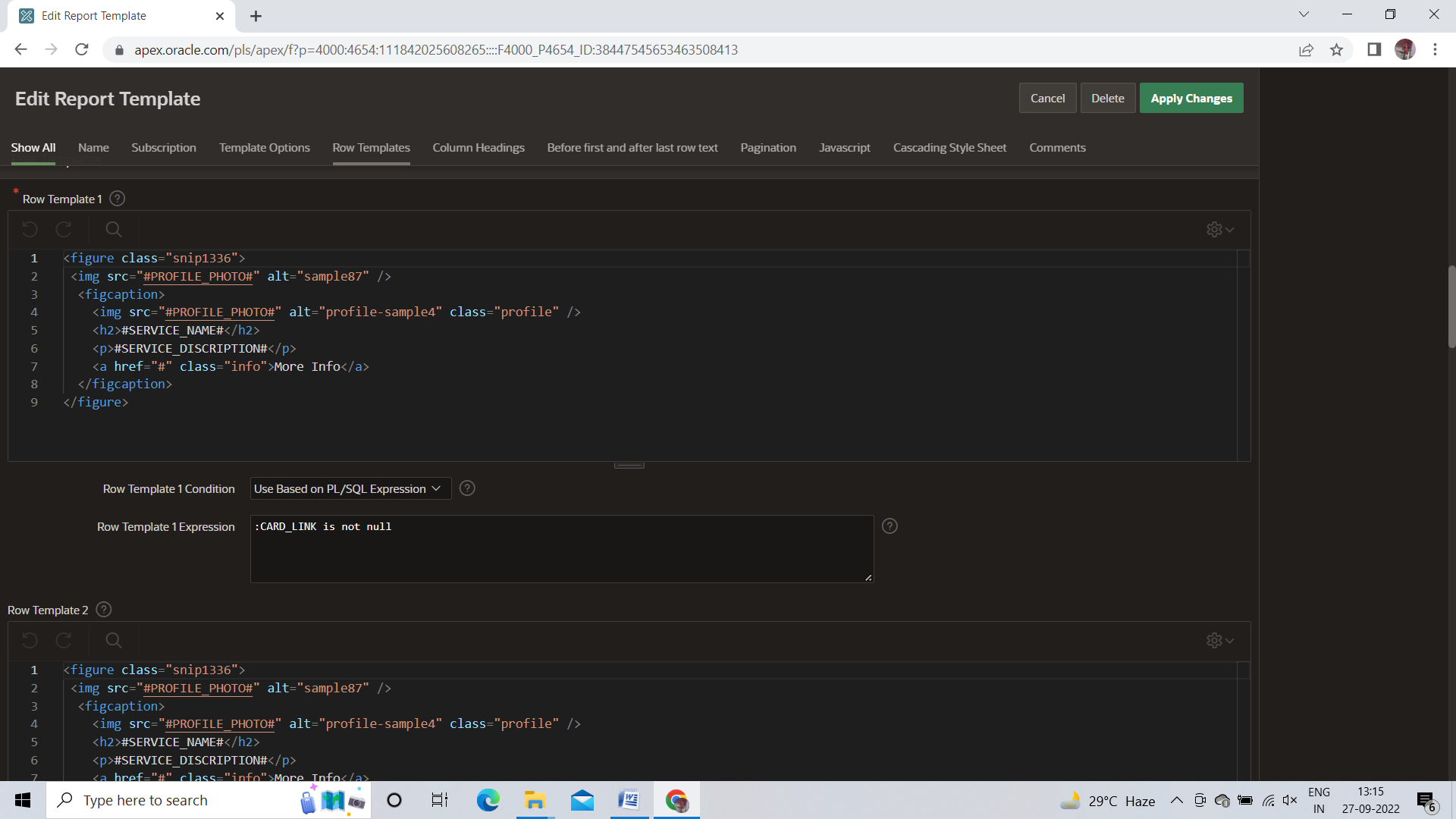
**Enter The Name of New Application and select as per your need then click on create Application**

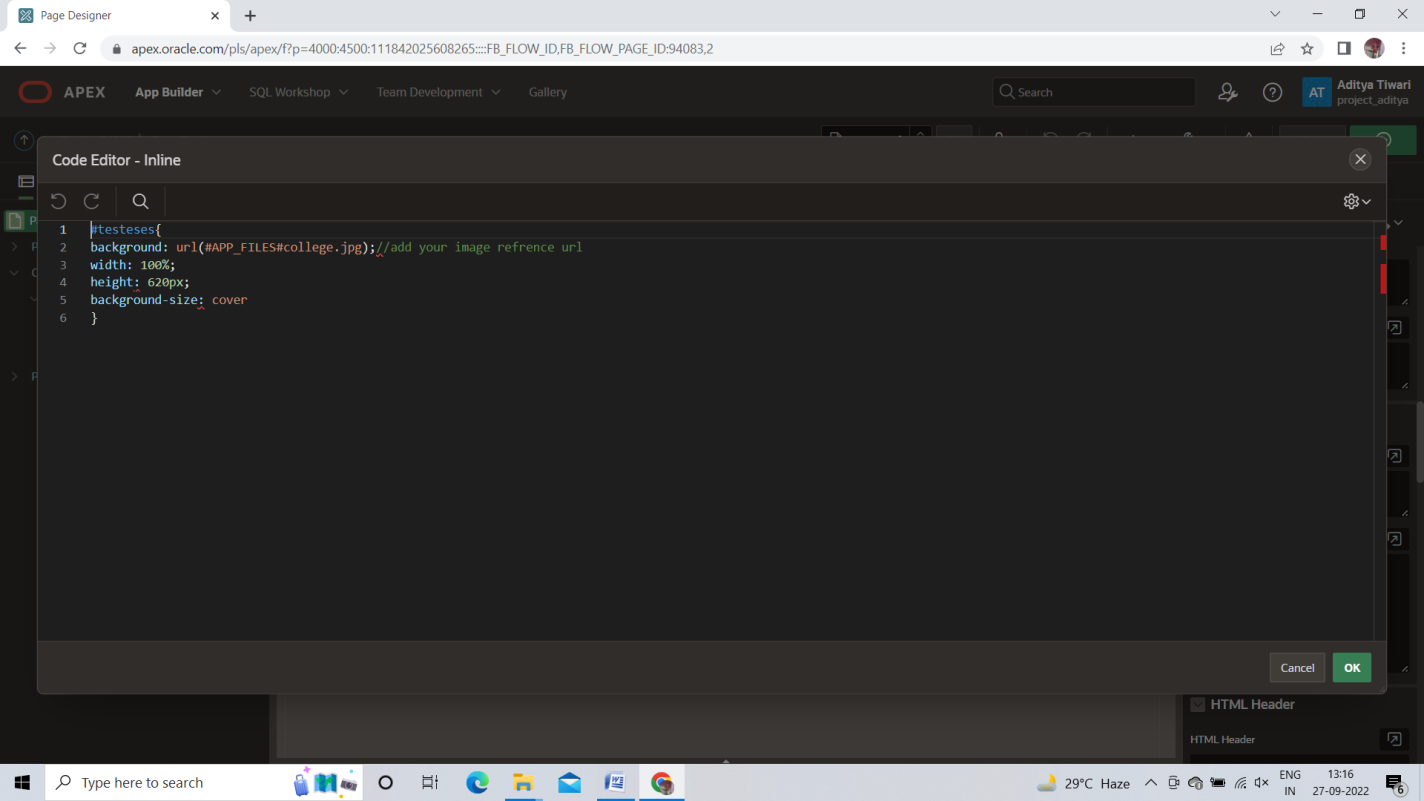


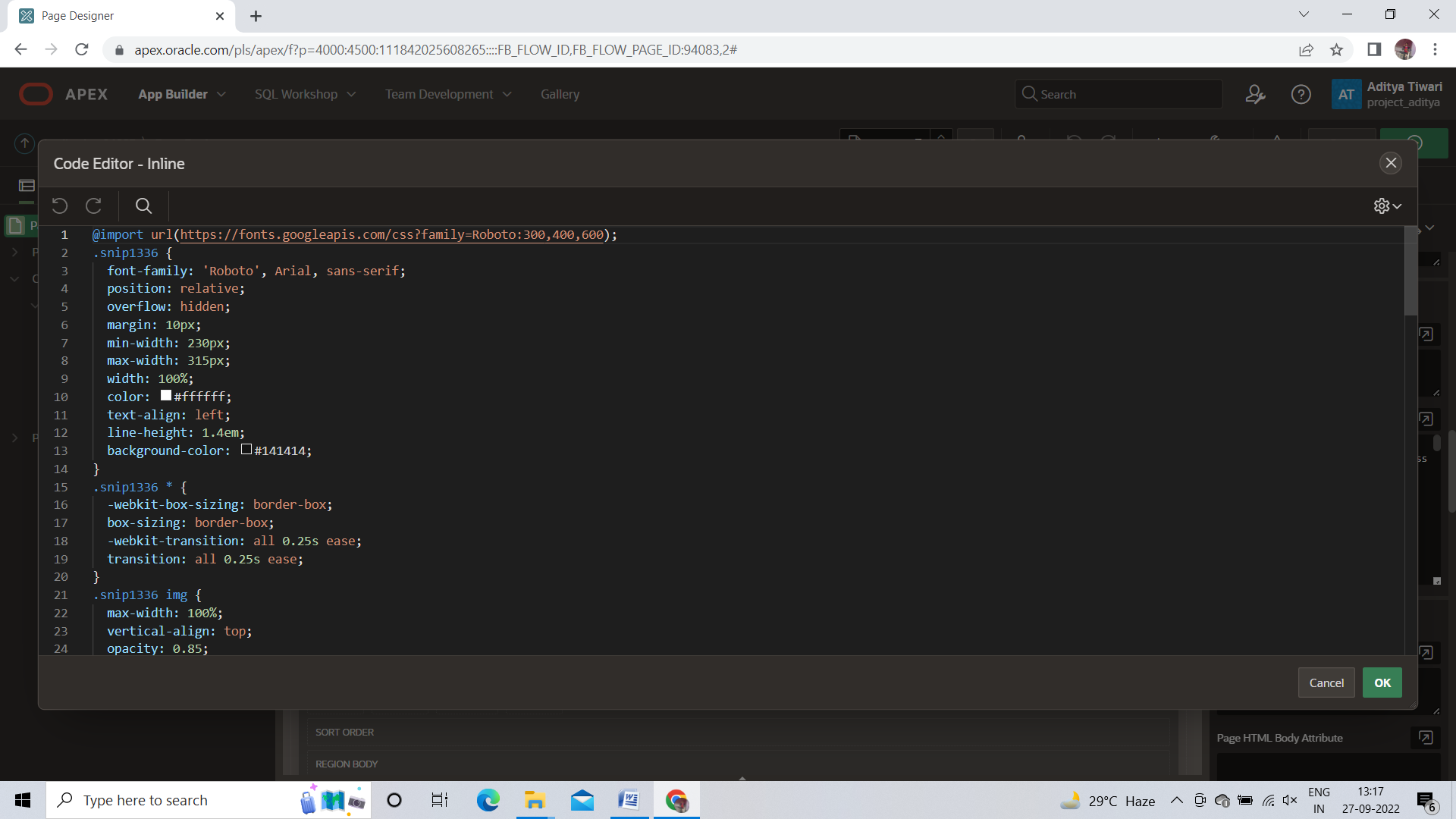
**Pages included in Inventory Management System:**

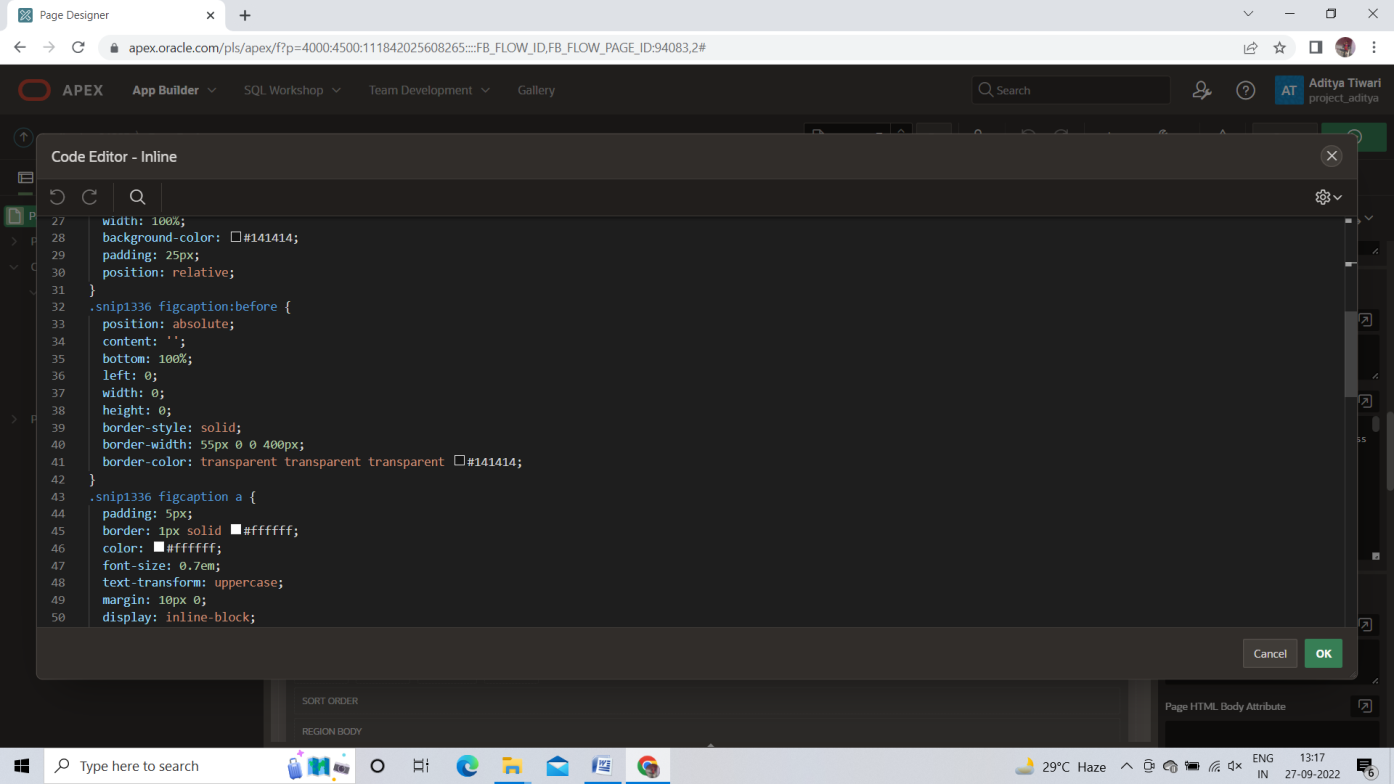
****

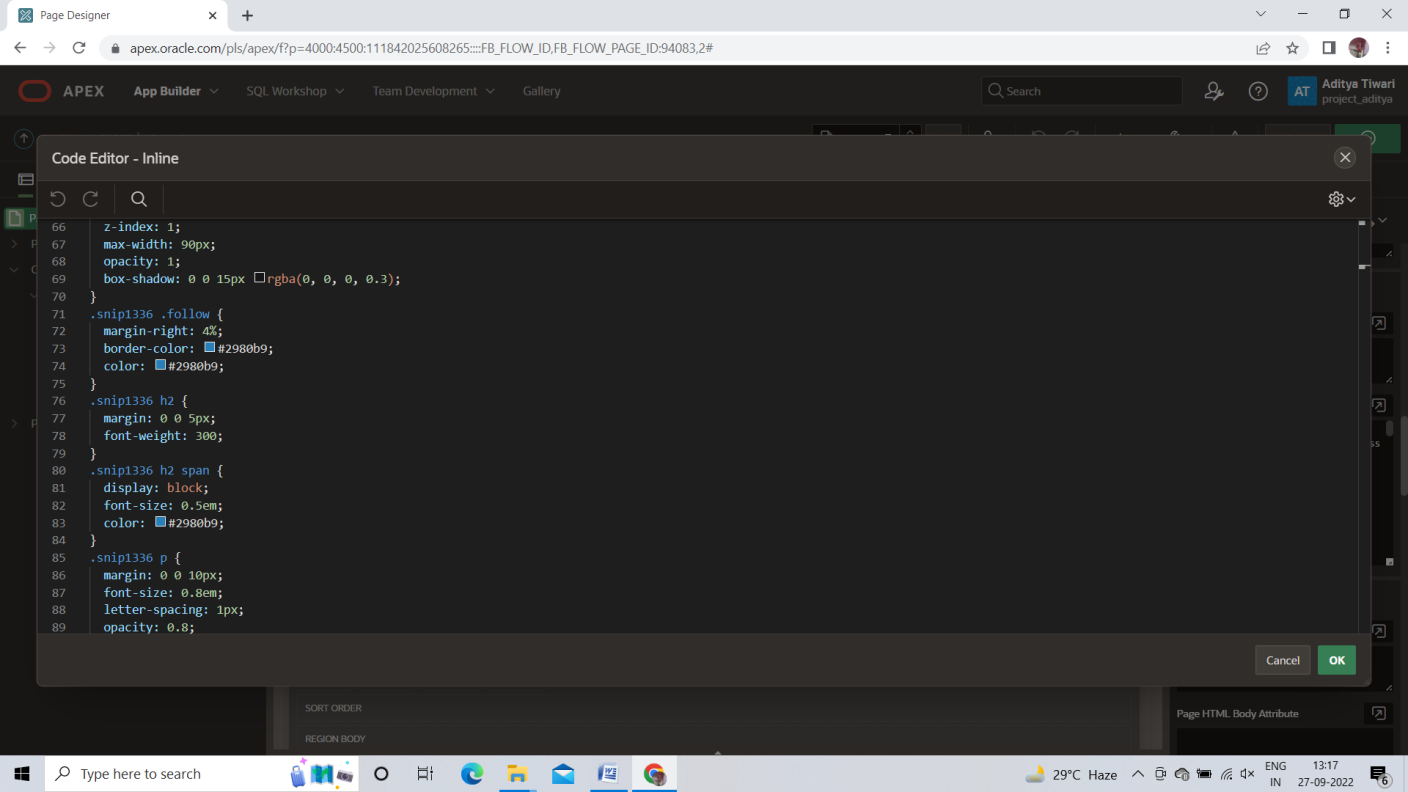
**Html, CSS and JavaScript:**

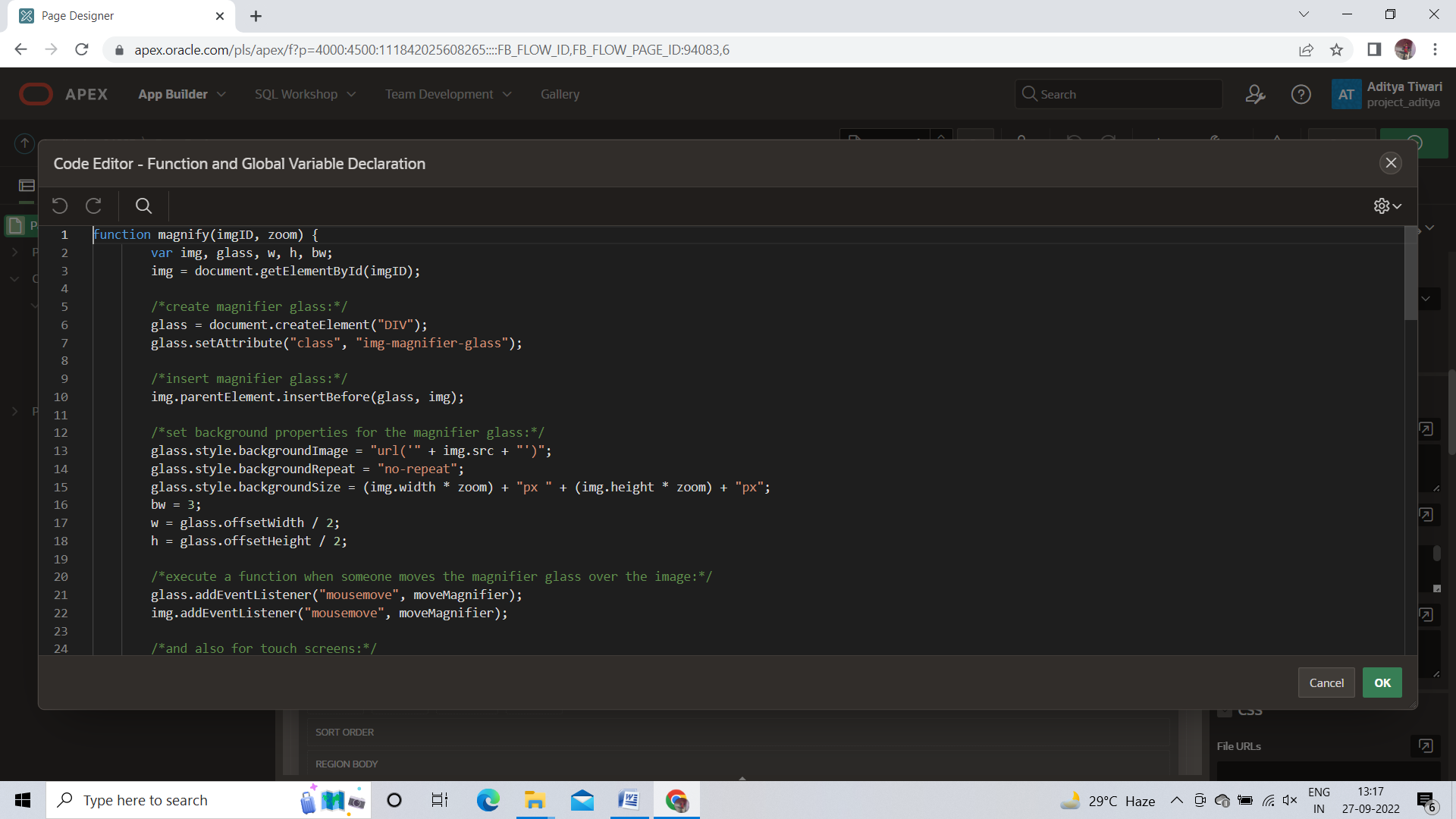
****

****

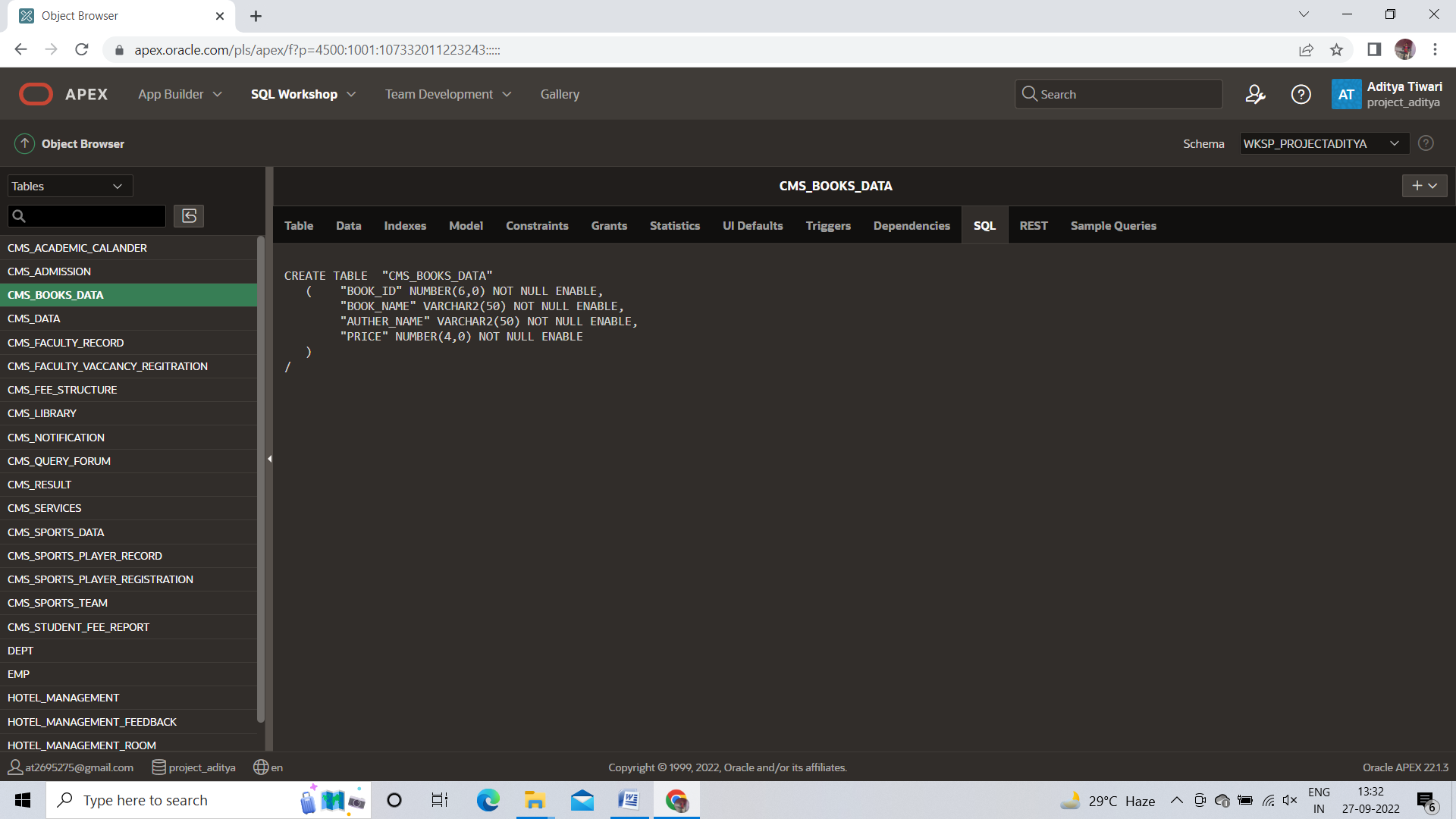


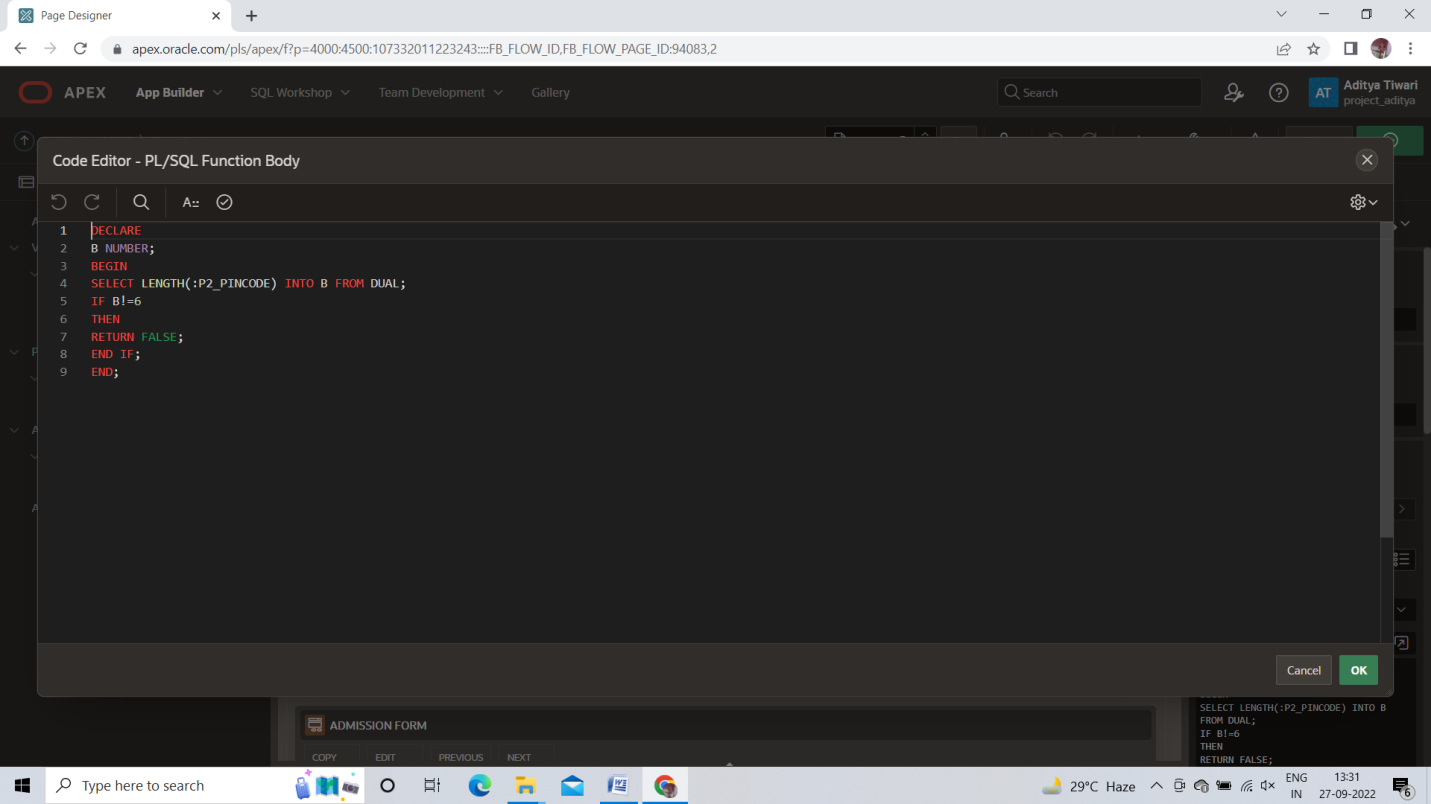






**Some SQL and PL/SQL Statements:**





**CHAPTER 5**

**RESULT & CONCLUSION**

Oracle Apex Application development is Rapid Application Development Tool Which is used to build a scalable, robust, portable and secured web applications instantly. It gives oracle database as cloud storage to store the data as per our need.

Oracle Apex is suitable for both coders and non coders but should have some basic knowledge about database. Apex contains thousands of functionality inside it, that results a more interactive and functionable web App.

These days, Enterprise Innovation focuses on improving business processes, driving efficiencies, enhancing customer experiences-all these are primarily driven by domain knowledge experts rather then technology experts. Adoption of low-code/no-code platforms and technologies allows diverse parts of the workforce, especially those with domain knowledge, to participate in the innovation process.

Most of The Web App Development companies moves to RAD (Rapid Application Development) Because an instant app development is much demanding right now. Each and every businessman and different department like college, banking department, e-commercial system needs to a rapid Application That can be easily manage and Low in cost. Oracle Apex is 20x faster to build and 100x less code due to this developer saves time and can do more interactive things within less time.

With Oracle Apex, we can easily build web application with essentially low knowledge of HTML, CSS, JavaScript. Oracle Apex is tightly integrated with Database and manages all of the processing associated with data manipulating in a database. If you have Basic Understanding with SQL, you can build beautiful, responsive web application very quickly. For data analysis, Oracle Apex is excellent as it supports the full SQL language, which is the Industry Standard for data Manipulation and aggregation.

Now, Oracle Apex is more Beneficial for Developers, Students, Business Users, Architect and CIOs that increases the demand of Oracle Apex continuously and Most of the Web Application Development companies moves towards Application Express (Apex)

**REFERENCES**

This Topic has been well written with the help of some books and online Journals submitted to various renowned organization as are referred below:

**Theory Book:**

* Oracle Apex 20 for beginner’s: By Riaz Ahmad.
* Oracle Application Express 19.1 Basics and Beyond: By Riaz Ahmad.
* Oracle 11g PL/SQL programming Workbook by Michael McLaughlin, Jahn M. Harper.

**Web References:**

* [www.oracle.com](http://www.oracle.com)
* [www.oracledevelopers.com](http://www.oracledevelopers.com)
* [www.W3schools.com](http://www.W3schools.com)
* [www.slideshare.net](http://www.slideshare.net)
* [www.riptutorial.com](http://www.riptutorial.com)
* [www.oracle.org](http://www.oracle.org)