**CHAPTER: 4**

**IMPLEMENTATIONS**

This chapter of the report describes the Functions, packages and modules used in the project:

**FRONT END**

**HTML:**

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

**CSS:**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

**Java Scripts:**

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it,and major web browsers have a dedicated JavaScript engine to execute it. As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has APIs for working with text, arrays, dates, regular expressions, and the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities. It relies upon the host environment in which it is embedded to provide these features.

**BACK END**

**Django Web Frameworks:**

Django is a Python-based free and open-source web framework, which follows the model-template-view (MTV) architectural pattern. It is maintained by the Django Software Foundation (DSF), an independent organization established as a 501(c)(3) non-profit.

Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

**Postgresql Database:**

PostgreSQL, also known as Postgres, is a free and open-source relational database management system (RDBMS) emphasizing extensibility and technical standards compliance. It is designed to handle a range of workloads, from single machines to data warehouses or Web services with many concurrent users. It is the default database for macOS Server, and is also available for Linux, FreeBSD, OpenBSD, and Windows.

**4.2 Functional Modules**

The functional modules included in the project are listed below:

**Insert Module:**

This module provides the functionality of collecting the required data from the designed interface and transmitting it to the appropriate table present in the database designed for this project. If the provided data does not satisfy the given constraints, it must refrain from storing it into the database.

**Update Module**:

This module again has the functionality of collecting the data from the designed interface, but it updates the already existing tuple that matches the provided primary key of the tuple to be updated, by replacing the existing attribute values with the newly collected data.

**Delete Module:**

The delete counterpart is loaded with the ability to delete a single or multiple records from the table. It searches for the tuple, in the query specified table, based on the provided value for an attribute. Admin can delete in the interface, based on which delete module searches for the record corresponding to that provided attribute value and deletes the record.

**Retrieve Module:**

The retrieve module has a basic functionality of accessing the entire specified table from the database and displays it.

**Trigger Module:**

Trigger in database is set of statements that are executed after an event occurs on the specified table. This is useful for logs wherein every change in database can be logged which helps keep a track of all changes/transactions on the database