Project/Training Objective

The objective behind the making of the EductaionBug was to make a single platform where students can get details about everything they need to know to choose a career.

They can get details about all the career options after 12^{th} in any stream.

It helps to know about different fields.

They can also check brief details about all the top colleges for Management, Engineering and Medical in India.

It not only helps the user to access it in a simple and attractive manner, but also helps Admin to handle the website very well.

The admin can control the list of top colleges using GUI.

Admin is also responsible to handle that which user should not have the access of using the website and who can access in User mode and Admin mode.

He can reply to all the comments/queries of the users through the email.

Brief of Training Company

OOPS INFO SOLUTION PVT LTD

(Manjit Singh, Ex-Infosys employee with 15 Year experience in IT, Java, J2EE, PHP, Android, Networking) Oops Info Solutions was established in 2003. His foundation and purpose is to provide and construct programs for existing companies and provide new and exciting updates to primitive bases technology. Oops Info Solutions Pvt. Ltd. is a India-based IT service provider offering services for technology infrastructure , enterprise resource planning, software application engineering and imparting Training. Oops was established on January 2003 and has consistently grown its client base. Featured in Computer Times, IT India and Computer World, With India as the base for centralized Project Management , Oops has expanded its representative office both in Chandigarh and Mohali in 2003. Oops also started operating software development center in the year 2004 and 2006. In 2011, Oops was nominated in "Spirit of Enterprise", which advocates and promotes entrepreneurial spirit of Indians. Oops is currently a registered member of Professional Training and Software Development. Oops is also ISO Certified Company. In 2013-14 Oops has also entered in Developing Mobile Apps. It is currently working in Designing mobile apps and integrating apps with the Websites.

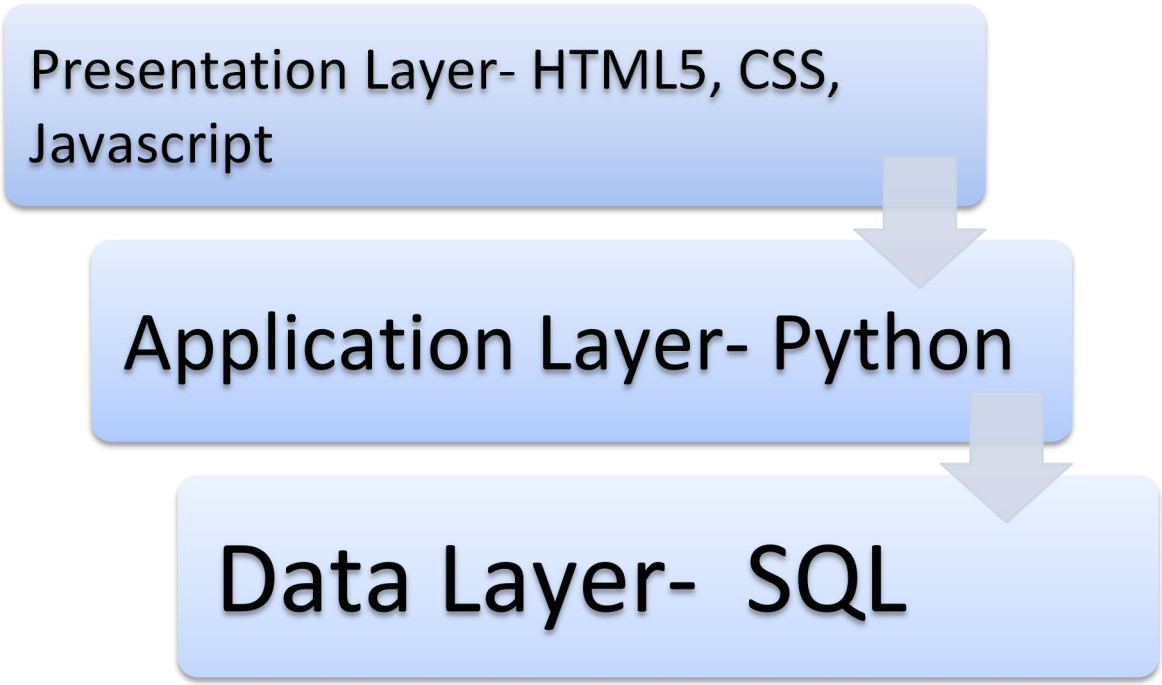
MODULAR DESCRIPTION OF THE JOB

The entire project has been made keeping the “three-tier architecture” in mind.

What is Three-tier architecture?

A 3-tier architecture is a type of software architecture which is composed of three “tiers” or “layers” of logical computing. They are often used in applications as a specific type of client-server system. 3-tier architectures provide many benefits for production and development environments by modularizing the user interface, business logic, and data storage layers. Doing so gives greater flexibility to development teams by allowing them to update a specific part of an application independently of the other parts. This added flexibility can improve overall time-to-market and decrease development cycle times by giving development teams the ability to replace or upgrade independent tiers without affecting the other parts of the system.

3-tier architectures consist of the following tiers:



Layers in three-tier architecture

1 PRESENTATION LAYER

The presentation tier is the front end layer in the 3-tier system and consists of the user interface. This user interface is often a graphical one accessible through a web browser or web-based application and which displays content and information useful to an end user. This tier is often built on web technologies such as HTML5, JavaScript, CSS, or through other popular web development frameworks, and communicates with others layers through API calls.

In this project, we have used HTML, CSS, Bootstrap, JavaScript.

The description will help to describe each and every module of the project.

1.1 HTML:

HTML stands for Hypertext Markup Language, and it is the most widely used language to write Web Pages.

- Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.
- As its name suggests, HTML is a Markup Language which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

In our Project, HTML was used to design a front end page which was used to navigate among the various tables and graphs which were created using GraphLab library for the collected data.

1.2 CSS:

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs,variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML. In our Project, CSS was used to beautify the front end page.

1.3 JAVASCRIPT:

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as LiveScript, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name LiveScript. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

In our setup, main function of javascript was to determine the current location at which the project is being ran.

1.4 BOOTSRAP:

Bootstrap is a free front-end framework for faster and easier web development. Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins. Bootstrap also gives you ability to easily create response designs.
Responsive web design is about creating web sites which automatically adjust themselves to look good on all devices, from small phones to large desktops.

Bootstrap 4 is the newest version of Bootstrap; with new components, faster stylesheet and more responsiveness.

Bootstrap 4 supports the latest, stable releases of all major browsers and platforms. However, Internet Explorer 9 and down is not supported.

Advantages of Bootstrap:

- Easy to use: Anybody with just basic knowledge of HTML and CSS can start using Bootstrap
- Responsive features: Bootstrap's responsive CSS adjusts to phones, tablets, and desktops
- Mobile-first approach: In Bootstrap, mobile-first styles are part of the core framework
- Browser compatibility: Bootstrap 4 is compatible with all modern browsers (Chrome, Firefox, Internet Explorer 10+, Safari, and Opera)

2 APPLICATION LAYER

The application tier contains the functional business logic which drives an application's core capabilities. It's often written in Java, .NET, C#, Python, C++, etc.

In this project, we have used python using Flask/Django Framework.

The description will help to describe each and every module of the project.

2.1 PYTHON:

In our setup of the project, Python helps us to integrate various libraries such as Flask, GraphLab, DateTime which is not easy to do in other languages.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

- Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Python's features include −

- Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- Easy-to-read − Python code is more clearly defined and visible to the eyes.
- Easy-to-maintain − Python's source code is fairly easy-to-maintain.
- A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- Extendable − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- Databases − Python provides interfaces to all major commercial databases.
- GUI Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- Scalable − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.

- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

2.2 FLASK:

Flask is a web application framework written in Python. It is developed by Armin Ronacher, who leads an international group of Python enthusiasts named Pocco. Flask is based on the Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

Flask is called a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more regularly than the core Flask program.

Flask is a layered architecture which has the flexibility to support different models of distribution and concurrency over the same data. The architecture eschews any fixed notion of concurrency control or any built- in model of distribution. Instead it provides a framework in which models can be defined and supported. One of the major difficulties in engineering such a system lies in the building of generic mechanisms to provide the facilities of data visibility restriction, stability, and atomicity, independently of the combination of these employed by a particular scheme.

The framework of the Flask architecture is shown in Figure 2 as a “V-shaped” layered architecture to signify the minimal functionality built-in at the lower layers. At the top layer the specifications of the model are used to guide the algorithms used to enforce them and can take advantage of the semantics of these algorithms to exploit potential concurrency [Gar83]. For example a particular specification may translate into an optimistic algorithm or alternatively a pessimistic one while the information they operate over remains the same. More importantly such an approach can accommodate different models of concurrency control and distribution.

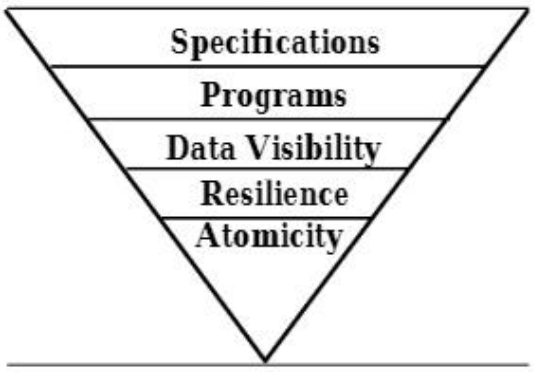


Fig: V-Shaped layered architecture

Flask is a generic layered architecture which is designed to provide sufficient flexibility to enable it to support many models. The Flask V-shaped structure builds in as little as possible at the lower layers, not only to provide the required flexibility, but also to free the lower layers from the burden of interference management and failure control. This greatly simplifies the construction requirements of these bottom layers and permits a variety of implementations to co-exist and be interchanged without the need for complete system rebuilds. Flask can thus be seen as a good platform for building, testing and comparing models of distribution and concurrency.

To demonstrate the feasibility and flexibility of this approach a persistent distributed application developed under the Flask architecture was described. The components of the instantiation involve three different hardware platforms, two object stores, two abstract machines, one communication and coherency protocol, and one concurrency control protocol running over four nodes.

2.3 DJANGO:

Django is a web development framework that assists in building and maintaining quality web applications. Django helps eliminate repetitive tasks making the development process an easy and time saving experience. Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Django makes it easier to build better web apps quickly and with less code.

Django is a registered trademark of the Django Software Foundation, and is licensed under BSD License.

History of Django

- 2003 − Started by Adrian Holovaty and Simon Willison as an internal project at the Lawrence Journal-World newspaper.
- 2005 − Released July 2005 and named it Django, after the jazz guitarist Django Reinhardt.
- 2005 − Mature enough to handle several high-traffic sites.
- Current − Django is now an open source project with contributors across the world.

Advantages of Django

- Object-Relational Mapping (ORM) Support − Django provides a bridge between the data model and the database engine, and supports a large set of database systems including MySQL, Oracle, Postgres, etc. Django also supports NoSQL database through Django-nonrel fork. For now, the only NoSQL databases supported are MongoDB and google app engine.
- Multilingual Support − Django supports multilingual websites through its built-in internationalization system. So you can develop your website, which would support multiple languages.
- Framework Support − Django has built-in support for Ajax, RSS, Caching and various other frameworks.
- Administration GUI − Django provides a nice ready-to-use user interface for administrative activities.
- Development Environment − Django comes with a lightweight web server to facilitate end-to-end application development

3 DATA LAYER

The data tier comprises of the database/data storage system and data access layer. Examples of such systems are MySQL, Oracle, PostgreSQL, Microsoft SQL Server, MongoDB, etc. Data is accessed by the application layer via API calls

.

In this project, we have used MySQL with WampServer.

The description will help to describe each and every module of the project.

3.1 DATABASE:

A relational database design was used to design the database. A relational database management system (RDBMS) is an excellent tool for organizing large amount of data and defining the relationship between the datasets in a consistent and understandable way. A RDBMS provides a structure which is flexible enough to accommodate almost any kind of data. Relationships between the tables were defined by creating special columns (keys), which contain the same set of values in each table. The tables can be joined in different combinations to extract the needed data. A RDBMS also offered flexibility that enabled redesign and regeneration of reports from the database without need to re-enter the data. Data dictionaries were used to provide definitions of the data used; these included the final data structures for the various tables and their corresponding data fields, description and sizes the user application programs and interface were developed using PHP, CSS, HTML, and Java Script with support of structured query language (SQL) and MYSQL.

RDBMS has become most predominant choice for the storage of information in new databases used for financial records, manufacturing, logistical data, personal data and much more since the 1980s. An RDBMS is a type of DBMS with a row-based table structure that connects related data

elements and includes functions that maintain the security, accuracy, integrity and consistency of the data.

Functions of relational database management systems

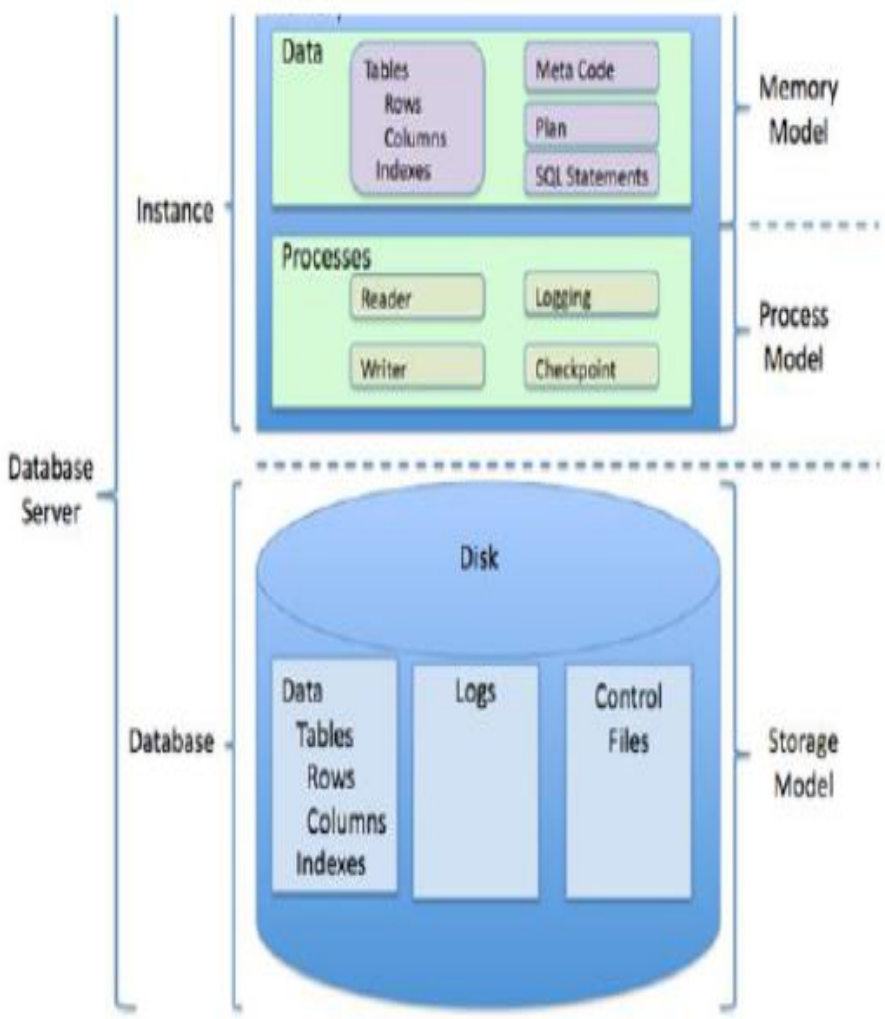
Elements of the relational database management system that overarch the basic relational database are so intrinsic to operations that it is hard to dissociate the two in practice.

The most basic RDBMS functions are related to create, read, update and delete operations, collectively known as CRUD They form the foundation of a well-organized system that promotes consistent treatment of data.

The RDBMS typically provides data dictionaries and metadata collections useful in data handling. These programmatically support well-defined data structures and relationships. Data storage management is a common capability of the RDBMS, and this has come to be defined by data objects that range from binary large object (blob) strings to stored procedures. Data objects like this extend the scope of basic relational database operations and can be handled in a variety of ways in different RDBMSes. The most common means of data access for the RDBMS is via SQL. Its main language components comprise data manipulation language (DML) and data definition language (DDL) statements. RDBMSes support the work of database administrators (DBAs) who must manage and monitor database activity. Utilities help automate data loading and database backup. RDBMSes manage log files that track system performance based on selected operational parameters. This enables measurement of database usage, capacity and performance, particularly query performance. RDBMSes provide graphical interfaces that help DBAs visualize database activity.

While not limited solely to the RDBMS, ACID compliance is an attribute of relational technology that has proved important in enterprise computing. Standing
for *atomicity*,*consistency*, *isolation* and *durability*, these capabilities have particularly suite

RDBMS for handling the transactions.



SQL is a language used to create, manipulate, examine and manage relational databases. SQL was standardized in 1992 so that a program could communicate with most database systems without having to change the SQL commands. Unfortunately one must connect to the database before sending SQL commands and each database vendor has a different interface as well as different extensions of SQL. Though SQL is well suited for manipulating database, it is unsuitable as a general application language and programmers use it primarily as a means of communicating with databases, another language is needed to feed SQL statements to a database and process results for visual display or report generation

WAMP server:

Databases are all around us. Almost every Blog, Website, E-mail services, E-commerce sites, and Cloud storage system needs a database to store data. Drupal, WordPress, Joomla, Ghost, os Commerce, OwnCloud, to name a few, need a database to store configuration and data.

If you are working on your computer with XAMPP or WAMPserver, or using cPanel to manage your website over Internet, it is almost inevitable that you will need to create a database.

Using WAMP server:

Run WAMP server as administrator. (Right-click on the WAMP icon found in the desktop or in the start menu, then select “Run as administrator. Click "Yes" on the "User Account Control" popup.)

After WAMP server has started, you will see the WAMP icon as shown below in the notification area at the bottom-right side of the desktop.

If the icon is green as shown above, all services like Apache, MySQL etc. are running. If the icon is not green, left-click on it. You will see the WAMP administrator panel as shown below. Click on “MySQL”. Then click MySql console from the drop down. Start using it for creating databases.