INSPIRATION OF FULL-STACK

----- A STORY------

Mr S.MOHAN., Mrs V.RADHA., Dr M.VARATHARAJ

Professors of Computer Science and Engineering Department

& Electrical and Electronics Engineering Department,

VSB College of Engineering Technical Campus, (An Autonomous Institution), Ealur Pirivu, Solavampalayam PO, Kinathukadavu, Coimbatore, Tamailnadu, India.

ABSTRACT :

We will discuss about Full-Stack which is come from the idea in database while it is stored an retrieve from the same .because it is full-stack .while to the both process of the store and retrieve of the data base and manipulate the any one language (query) that is request and response of the database first it is activated as sql (oracle).then come to MySQL and then now used NoSQL which is also called DBMS or RDBMS .then BIGDATA these are all ideas comes from the sql. The full stack developer utilize both query language and frame work . the frame work provide structure of tool for building both the front end and back end of an application. Write a query language are necessary for interacting with the database to image application data so we will create a database is either store or retrieves with the help of query languages sql. separating. now a day we will use both technology with one query languages or frame work languages suited to the project depends on the requirements and specific technology stack performance.

KEYWORDS - RDBMS, DBMS, SQl, MySQL & NoSQL

Introduction:

In query languages used to retrieval data and store the data to the database suitable sql query . that query used to depends upon the situation it is either create and stored and manipulate or retrieve the data from the database in the suitable query from the query languages that query is asked from the database then response data from the database which is act as front end and back end of the full stack.

The front end means store the data in to the data base. But in back end means retrieve data from the database the front end are used query command and back end used to another query both query are used to sql. because these queries are used to inspiration of full stack the full stack is the request and response or converted in to the front end and back end the separate languages merge with the one language . that is full stack that languages are spring boot in java, python , java script (MERN)and (MEAN) are also used with the full stack coding in the trend technology. So, we will discuss about it.,

Core Components

In full stack data getting from database via through the backend data giving to the database via through the frontend In the both process are called full-stack it is idea comes from SQL because SQL is the core component of full-stack SQL is the query language because it is asking questions and getting related answer from respective database so it is called request and response this database either frontend or backend because it will do user need the core components are used to store and retrieve manipulate the data from the database through the SQL the fundamental structure of the SQL is the idea of core component to the full-stack while SQL is not directly used to on the frontend and data display and manipulate in the UI originate from the database and the frontend send request to the backend to react with the data . The backend is where SQL plays a unique

role It execute query to interact with the database retrieving inserting updating and deleting database on the application logic and use request .

Database-Management

This layer invoking storing and managing application data efficiently relational database which relay heavily an SQL one common choice example: My-SQL, Postgre SQL, SQL-server and SQLite SQL is the fundamental of full-stack developer an working with relational database enabling to the design database schema store and retrieve data ensure data integrating and upgrading full performance.

Which a frontend provide the user interface and back-end handles a application logic, SQL is the essential language that enables backend efficiently manage and interact with database stored in relational database within a full stack application.

Cod for Full stack from SQL

Create table user(id int, auto increment primary key, name varchar(10) not null, create at time stamp default current- timestamp);

Front end or client –side logic

Html,css,java script

<!doc type html>

<html>

<head><title>under input</title>

</head><body>

<Hi> enter under data</h1>

<from id=”user data form>

<label for=”name”>id=”name>

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

<button type=”submit”>submit</label?

</form>

API application program interface how to interface

Front end and back end communication

Example

(node.js,express.js,sever.js)

Interface back end

Const express=require (‘express’)

Const app=express()

Const point=3000;

App. use(express. Json());

(will connect to the sql)

Assuming the database connection modules)

Const app =required (./database');

App. post('/api/user/,any one(req,res))

{

Const [name]= request body

Try

{

(insert into database)

Const=awaiting database query

('insert into the user(name) vlues((?);(name));

Res.status(200).json)(id:reslt,insert id,name)

}}

Catch(error)

{console.error

{console .error(error creating name,error);

Res.status(500).json(error',failed to create user');

Create user});

}

App.listern(point,()=consol.log(server testing at

http:/local host:${point});

the front end request(example form submission) to te back end API end point ,the back end process those request ,interact with the sql database to store or retrieve data and send response back to the front end which update UI accordingly.

In sql needs to server output on

Code

Set server output on;

Declare

v-message varchar2(100)

begin

v-message:="hello" from pl/sql;

dbms-output.put-line('thi is test massage');

dbms-output.put-line(‘the variable constraint);||v-message);

declare

v-sum number;

begin

v-sum:=10+20;

dbms-output.put-line(‘the sum is’||v-sum);

end;end;

output

tis test message:

the variable constraint:hello frompl/sql;the sum is=30.

Comparison

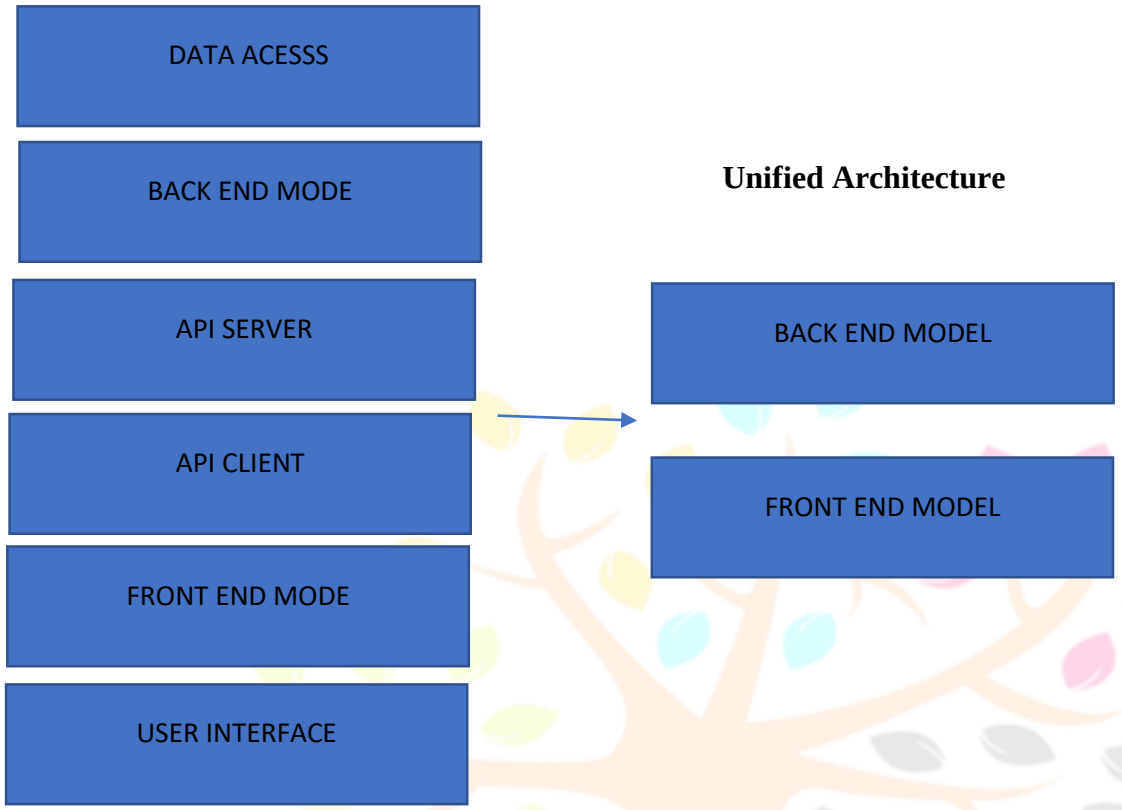
SQL(structure query language).the full stack development represent distinct but inter connected aspect if the software development. SQL the specific language design for managing and manipulating relational database it primary function is to create, retrieve, update and delete data with in the data structure database. SQL used various role with in the software developments inter changing database administrator back end develop data analysis and even full stack developer.

Full stack development encourage entire spectrum of the web development from the value facing front end to the server side back end underling database in full stack developer capable of working on all layer of an application interchanging. Sql is the control tool with in the full stack developing tool kit particularly for the application utilize relational database like mysql, poatgresql or sql sever a full stack developer need to understand how to design database write efficient sql query and integrated database instruction in to back end logic.

Sql is the specialized language for database instruction when as a full stack development is a comparative approach to building entire software development application often incorporating sql as a traditional element for data interchanges.

Architecture of full-stack application

Traditional architecture



Conclution

In this paper shows the basic idea comes from the sq query. because query means questions. That is asking question means get related answer from the database, the principle also like as request and response data request from the database and then database gives the response to the suitable data(answer). Nowadays we will use full stack that is front end and back end .the front end means client side application the back end means server side logic this is the logic based by rhe query fiction. Because it is the request and response from the database to the database. So we will tell the story from the sql to the full stack idea the sql is the core component of the full stack. The inspiration of full stack ideas comes from the sql.

References

1. S. K. Bajracharya and C. V. Lopes, “Analyzing and mining a code search engine usage log,” *Empirical Softw. Eng.*, vol. 17, no. 4/5, pp. 424–466, 2012.

2. S. K. Bajracharya, J. Ossher, and C. V. Lopes, “Leveraging usage similarity for effective retrieval of examples in code repositories,” in *Proc. 8th ACM SIGSOFT Int. Symp. Found. Softw. Eng.*, 2010, pp. 157–166.

3. M. G. Bulmer, *Principles of Statistics*. North Chelmsford, MA, USA : Courier Corporation, 1967.

4. C. Carpineto and G. Romano, “A survey of automatic query expansion in information retrieval,” *ACM Comput. Surveys*, vol. 44, pp. 1–56, 2012.

5. S. Chatterjee, S. Juvekar, and K. Sen, “Sniff: A search engine for java using free-form queries,” in *Proc. 12th Int. Conf. Fundam. Approaches Softw. Eng.*, 2009, pp. 385–400.

6. R. Cox, “Regular expression matching with a trigram index or how google code search worked,” 2012. [Online] Resource: http://swtch.com/rsc/regexp/regexp4.html

7. CreativeResearchSystems, Sample Size Calculator, 2012. [Online]. Available: http://www.surveysystem.com/sscalc.htm, Accessed on: 18 - Feb. 2016.

8. L. R. Dice, “Measures of the amount of ecologic association between species,” *Ecology*, vol. 26, pp. 297–302, 2012.

9. G. Gay, S. Haiduc, A. Marcus, and T. Menzies, “On the use of relevance feedback in IR-based concept location,” in *Proc. Int. Conf. Softw. Maintenance*, 2009, pp. 997–1016.

10. X. Ge, D. C. Shepherd, K. Damevski, and E. R. Murphy-Hill, “How developers use multi-recommendation system in local code search,” in *Proc. Int. Conf. Visual Language Human-Centric Comput.*, 2014, pp. 69–76.

11. S. Gottipati, D. Lo, and J. Jiang, “Finding relevant answers in software forums,” in *Proc. 26th Int. Conf. Automated Softw. Eng.*, 2011, pp. 323–332.

12. M. Grechanik, C. Fu, Q. Xie, C. McMillan, D. Poshyvanyk, and C. Cumby, “Exemplar: A source code search engine for finding highly relevant applications,” *IEEE Trans. Softw. Eng.*, vol. 38, no. 5, pp. 1069–1087, Sep. -Oct. 2012.

13. P. E. Greenwood and M. S. Nikulin, *A Guide to Chi-Squared Testing*. New York, NY, USA : Wiley, 1996.

14. S. Haiduc, G. Bavota, A. Marcus, R. Oliveto, A. D. Lucia, and T. Menzies, “Automatic query reformulations for textual retireval in software engineeing,” in *Proc. Int. Conf. Softw. Eng.*, 2013, pp. 842–851.

15. J. H. Hayes, A. Dekhtyar, and S. K. Sundaram, “Advancing candidate link generation for requirements tracing: The study of methods,” *IEEE Trans. Softw. Eng.*, vol. 32, no. 1, pp. 4–19, Jan. 2006.