

DEVELOPMENT OF SECURE INFORMATION MANAGEMENT SYSTEM USING SINGLE PAGE APPLICATION APPROACH

Akshay M S¹, Akshit Dhar², Dhanush Iyer Y R³, Hari Prasad C N⁴, Dr. Rajiv Ratan Chetwani⁵, Dr. Balakrishna R⁶,

¹Dept. of ISE, RRCE, Bangalore, India

²Dept. of ISE, RRCE, Bangalore, India

³Dept. of ISE, RRCE, Bangalore, India

⁴Dept. of ISPO, ISRO Headquarters, Bangalore, India

⁵Director, Dept. of ISPO, ISRO Headquarters, Bangalore, India

⁶Principal, RRCE, Bangalore, India

Abstract— Information system provides various details about various aspects of the system such as resources, assets which help to make an efficient analysis and to take better decisions. To develop an application, various security measures have been taken like authentication, validations, session management and also the development of single page web application by using the multi-tier web architecture which offers high performance and many perks over the standard multi-page web application. Manpower Information deals with information regarding employees in an organization. Our paper focuses on developing SPA for handling this manpower information at a large scale. We have used ExtJS on the client-side and servlets on the server side with MongoDB as the no-SQL type database.

Index Terms— SPA, NOSQL, MongoDB, ExtJS.

I. INTRODUCTION

Information growing exponentially in the digital world and varies from one organization/department to another. This information shall be related to asset, resources, manpower etc., requires an efficient information system to manage such data. In this paper we are presenting a use case of design and development of manpower Information management system to manage the information regarding employee profile, skill sets, memberships, assets, joining and retiring dates, awards, achievements etc. Secure coding practices were followed during the development. Manpower information deals with information of the employees of the organization. This information once exploited can play a huge role in understanding the organization's workforce and provide significant insights into them. This information helps to build analytics of the employee profile to take some decision. The software was designed using Single Page Application (SPA) development approach to host on the web. There is hundreds of framework out there which supports the development of SPA but ExtJS, a JavaScript framework, stands apart with its rich library of interactive and visually appealing components. It is built as an add-on library with the inclusion of interoperability

with JQuery and prototype. It provides cross-platform browser compatibility and is excellent for managing and handling information through its grid components. Various security measures can be incorporated into SPA such as encryption, DBA authentication, and IP authentication. Encryption plays an important factor in ensuring security by encrypting the parameters passed to the server. DBA authentication ensures the security by providing the credentials to access the application. IP authentication ensures security by blocking all the unauthorized IP addresses i.e., IP's other than that of an organization.

MongoDB is used to store the data, which is an open source Nosql type database wherein it stands for "not only SQL". Since it's not relational, it doesn't store data in the table form rather; data is stored in the key-value pair. These Nosql type databases are schema-less which reduces the complexity and also stores any kind of data which may be structured, unstructured or semi-structured. From analysis and with various testing MongoDB proved it resists the SQL injection.

II. METHODOLOGY

In our project we have used the technique of full stack development, where to avoid complexity and to use the codes or data redundantly and efficiently. [6] Model-View-Control (MVC) architecture was followed with clear distinction between client and server side implementations.

An ExtJS framework used to design and develop Single Page Web App for all client side modules – Login, Profile view, add, modify. And all server side modules (business logic) were developed using java, servlets and open source libraries.

The architecture on which we have implemented is multi-tier web architecture.

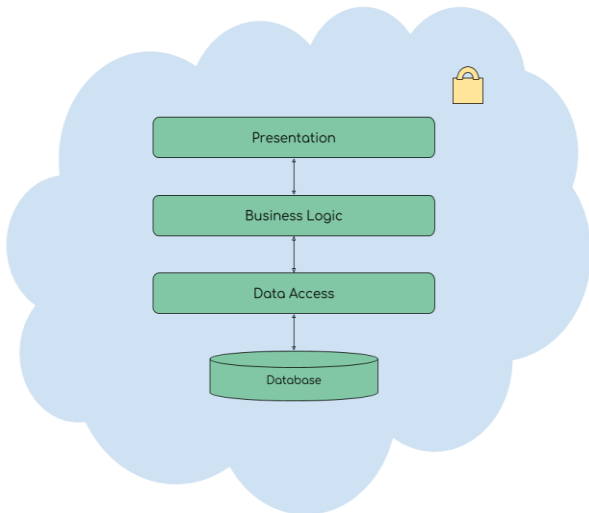


Figure 1: Multi-tier web architecture

From the figure 1, the multi-tier web architecture we have three consecutive layers. The first layer is the presentation/application layer; the second layer is connection oriented layer where we are implementing a Java servlets. The third layer is the database layer where the data are stored.

In the first layer the Ext Js, the JavaScript framework plays an important role to create a single page application, which has the beautiful user interface UI components and provides a high performance data exchange and data sharing feature. The design will limit the application structure to fit into only a single html page and also it compresses the navigational capabilities. This kind of structure will increase the performances of the application and also makes the development easier. This application will manipulate the data and also will control the flow in a single page which more effective and makes web application the feature rich and content rich application.

In the second layer, which is an application layer, web server will be serving all user requests. All modules were bundled into web archive and deployed in the web server.

To address the dynamic schema, and to store unstructured data, Nosql database was found suitable. The advantage of Nosql is that it supports the nesting levels at maximum level of the documents redundant operations will not matter in these kinds of databases. If these techniques are practiced then the database performance will be high.

This web application is been deployed in the secured cloud, where unauthorized end-users cannot access the application.

Algorithm

- Step 1:** Create a login Page, for authentication.
- Step 2:** Perform the security checks.
- Step 3:** If authentication fails, print invalid credentials.
- Step 4:** If authentication succeeds, dashboard appears which describes the analytical part of the data entered.

- Step 5:** Home page exists which it consists of minimal information in a grid panel.
- Step 6:** Provide an update option to update the data.
- Step 7:** If required, go to insertion form fill the data.
- Step 8:** If new user, ask to fill the data compulsorily.
- Step 9:** When the data is filled in the insertion form, update the home page.
- Step 10:** Logout

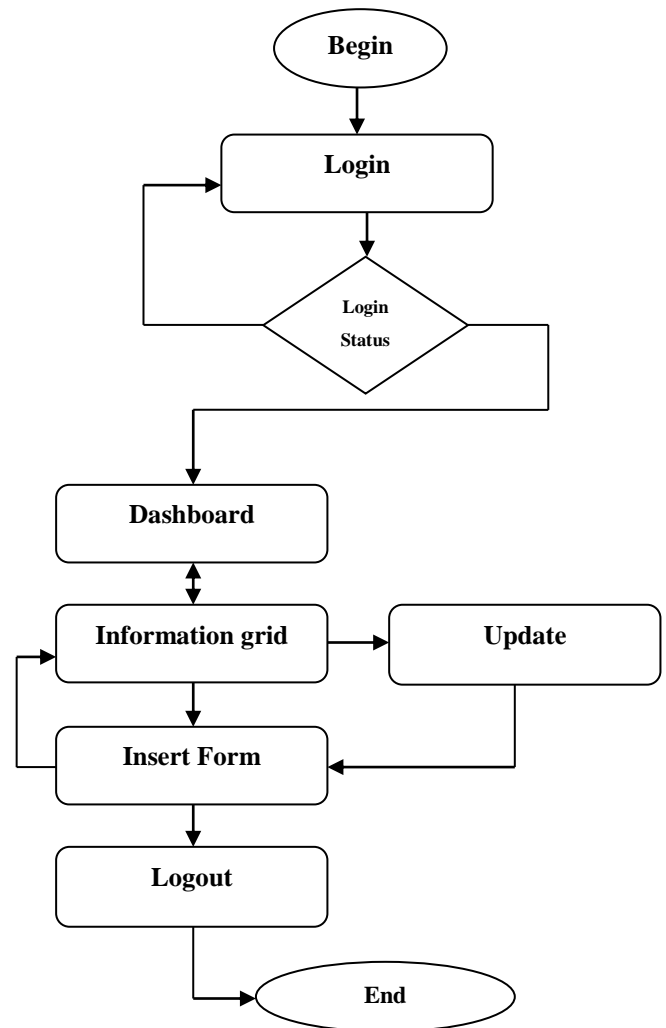


Figure 2: Flow Chart of the Application

III. SECURE CODING PRACTICES

A. Encryption to provide security

All users can access the application by using login credentials which are given by the DBA. Once the user gets authorization from the admin he or she is allowed to login to the application and perform various tasks. All data received at the server are validated at the server. The login password information is stored in encrypted form. These server side validations prevent the middleman attack.

B. Validation at server side to prevent malicious

In this information era, no web application is secured. The confidential information shared between the web applications may be vulnerable to various attacks. The database is the target for the attackers to modify the information. To get rid of malicious attacks validation is being performed on all the data in every request, where data cannot be miss-interrupted. In addition to server side validation, client side validation also performed on each and every data. Any user may enter invalid data to cause the injection flaw. It is very difficult to identify the bug while testing. A small snippet may enter by the unauthorized user, which leads to the data misinterpretations. To prevent the injections the best option to use of white list input validation.

C. Session management

A Session a timeline between entry and exit (login to logout) will be created on successful login authentication. Each session has a unique session id which will be used to identify the session. Session id also is used to keep track of the authorized users from the beginning of the request. The login authentication guarantees the capacity to distinguish the user on any consequent requests and also having the capacity to apply security access controls, approved access to the client private information, and to expand the convenience of the application. It prevents any unauthenticated users from gaining access to the web application. The timestamp may be grouped with the session ID and then stored in a hash form thus providing another level of security. This prevents the attackers to access the session ID. If redundant attempts are made on the encrypted session ID, the user will be not allowed to login to the application.

IV. RESULTS AND DISCUSSION

The outcome is a Single page web application, which consists of the various heterogeneous data in a single system say centralized system. This manpower information allows performing operations on it and generating useful insights on it.

This cloud based web application ensures that the information is accessible within the scope of organization and is not vulnerable to any malicious attack.

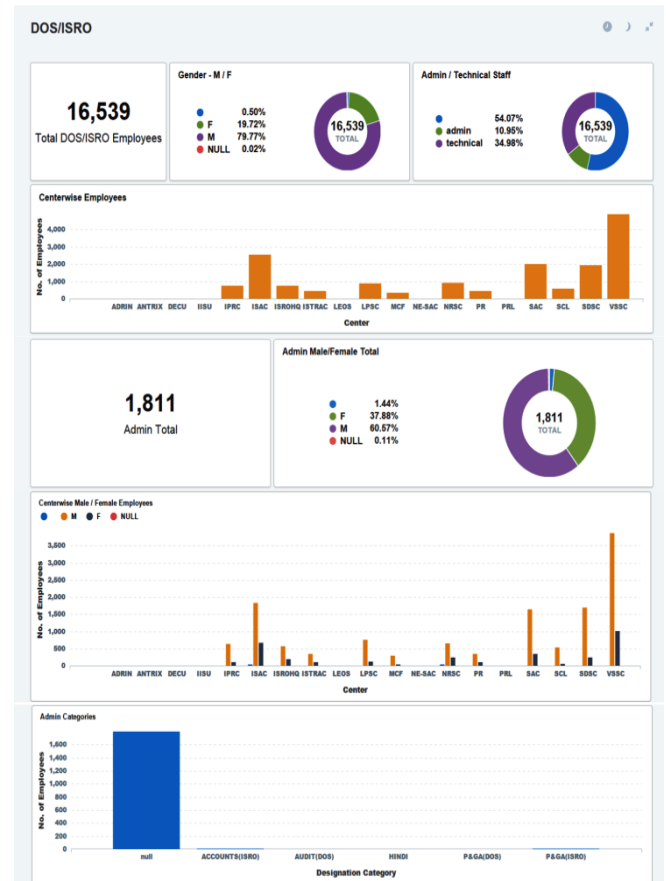


Figure 3: Dashboard analytics

Figure 3 illustrates the analytical part regarding the employee details. By viewing this analytics some decision can be take or made.

Functions	Staff Code	First Name	Last Name	Gender	Email	Mobile Number
	12437893	Aksh	Dhar	male	akshdhar@tve.com	720400915
	78451234	Akshay	MS	male	akshay.mvivas.as4@gmail.com	784503211
	123456789	Mahyn	Monroe	female	myn@vva.com	8807546512
	159487263	Dhanush	Iyer	male	iyerdhanush@gmail.com	7986532144

Figure 4: Data Grid

Figure 4 illustrates the list of employees with their minimal information and also with various menu options.

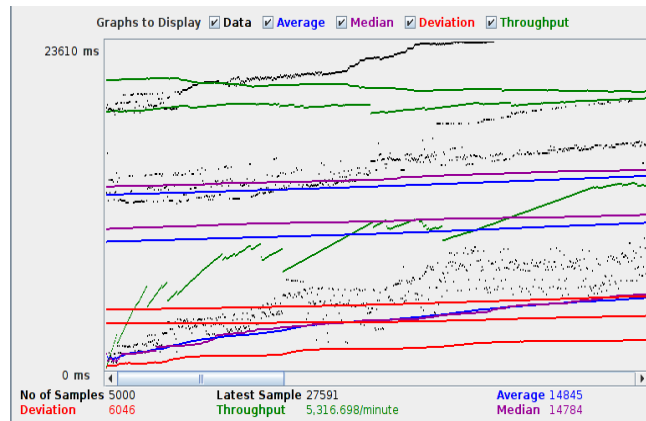


Figure 5: Test Simulation Graph

Figure 4.3 illustrates the efficiency of the application considering different feature like deviation, throughput and median. From the figure, it is clear that the application runs efficiently.

V. CONCLUSION

We have proposed the solution which restructures the traditional web application to single page application. Our team has explored various techniques, design patterns and methodology to implement the software. This software developed, deployed and works as a centralized information system with analytics. By considering manpower information in an organization in different data centers, we have collected the data in offline and populated in the DB. Synchronization with all databases located at different locations was proposed and accepted.

VI. ACKNOWLEDGEMENT

Heartfelt thanks to ISRO for providing the opportunity learn and work with esteemed analysts and scientists in the department of Information Systems Program Organization (ISPO). "Director of ISPO" Shri. Rajiv Ratan Chetwani for giving us continuous support and encouragement in completion of the project. We would also like to thank Mr. Hari Prasad C N for his guidance and support. A special thanks to project coordinator at Rajarajeswari College of Engineering and our internal guide Dr. Balakrishna R, Principal, RRCE.

REFERENCES

- [1] Hamouda, S. and Zainol, Z., 2017, August. Document-Oriented Data Schema for Relational Database Migration to NoSQL. In Big Data Innovations and Applications (Innovate-Data), 2017 International Conference on (pp. 43-50). IEEE
- [2] Kumar, J. and Garg, V., 2017, October. Security analysis of unstructured data in NOSQL MongoDB database. In Computing And Communication Technologies for Smart Nation (IC3TSN), 2017 International Conference on (pp. 300-305). IEEE.
- [3] Kurdi, R. and Randles, M., 2013, December. Proposed Design for an E-Government Web Portal Application Using Cloud Computing. In Developments in eSystems Engineering (DeSE), 2013 Sixth International Conference on (pp. 317-322). IEEE.
- [4] Li, B., Zhu, Y., Sun, Y., Grace, A. and lay Teo, K., 2018. Deterministic conversion of uncertain manpower planning optimization problem. IEEE Transactions on Fuzzy Systems.
- [5] Maraj, A., Rogova, E., Jakupi, G. and Grajqevci, X., 2017, October. Testing techniques and analysis of SQL injection attacks. In Knowledge Engineering and Applications (ICKEA), 2017 2nd International Conference on (pp. 55-59). IEEE.
- [6] Oh, J., Ahn, W.H. and Kim, T., 2016, August. MVC architecture driven restructuring to achieve client-side web page composition. In Software Engineering and Service Science (ICSESS), 2016 7th IEEE International Conference on (pp. 45-53). IEEE.
- [7] Tesarik, J., Dolezal, L. and Kollmann, C., 2008, August. User interface design practices in simple single page web applications. In Applications of Digital Information and Web Technologies, 2008. ICADIWT 2008. First International Conference on the (pp. 223-228). IEEE.
- [8] Vela, B., Cavero, J.M., Cáceres, P., Sierra, A. and Cuesta, C.E., 2017, June. Defining a NoSQL Document Database of Accessible Transport Routes. In Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2017 IEEE International Conference on (pp. 1125-1129). IEEE.
- [9] Zhang, W., Zhang, H., Chen, H., Zhang, Q. and Cheng, A.M., 2012, May. Improving the QoS of web applications across multiple virtual machines in cloud computing environment. In Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International (pp. 2247-2253). IEEE.