**XML BASED DATABASE MIGRATION**

##### A PROJECT REPORT

###### ***Submitted by***

**KUSMA THUMMAGUNTA**

**SHAMA HINA.D**

**Under the guidance of**

**Mrs. J. BRINDHA MERIN, AP(Sr.G)/CSE**

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

**BACHELOR OF TECHNOLOGY**

*in*

# COMPUTER SCIENCE AND ENGINEERING



## B.S.ABDUR RAHMAN CRESCENT INSTITUTE OFF SCIENCE & TECHNOLOGY

(Estd. u/s 3 of the UGC Act. 1956)

https://crescent.education/

##### July 2020



## B.S.ABDUR RAHMAN CRESCENT INSTITUTE OF SCIENCE & TECHNOLOGY

(Estd. u/s 3 of the UGC Act. 1956)

https://crescent.education/

**BONAFIDE CERTIFICATE**

Certified that this project report “XML BASED DATABASE MIGRATION” is the bonafide work of “KUSMA THUMMAGUNTA(160071601171) and SHAMA HINA (160071601132)” who carried out the project work under my supervision. Certified further, that to the best of our knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

*SIGNATURE* *SIGNATURE*

**Mrs. J. BRINDHA MERIN Dr.E. SYED MOHAMED**

**SUPERVISOR HEAD OF THE DEPARTMENT**

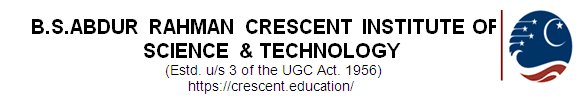
Assistant Professor (Sr. grade) Professor & Head

Department of CSE Department of CSE

B.S. Abdur Rahman Crescent B.S. Abdur Rahman Crescent

Institute of Science and Technology Institute of Science and Technology

Vandalur, Chennai – 600 048 Vandalur, Chennai – 600 048



**VIVA VOCE EXAMINATION**

The viva voce examination of the project work titled **“XML BASED DATABASE MIGRATION”,** submitted by **KUSMA THUMMAGUNTA (160071601171) and SHAMA HINA (160071601132)** is held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

We sincerely express our heartfelt gratitude to **Dr. A.PEER MOHAMED,** ProVice Chancellor, B.S. Abdur Rahman Crescent Institute of Science and Technology,for providing us an environment to carry out our course successfully.

We sincerely thank **Dr. A. AZAD**, Registrar, for furnishing every essential facility for doing our project.

We thank **Dr. VENKATESAN SELVAM**, Professor and Dean (SCIMS) and **Dr. E. SYED MOHAMED**, Professor and Head, Department of Computer Science and Engineering, for providing strong oversight of vision, strategic direction and valuable suggestions.

We obliged to our internal guide **Mrs. J. BRINDHA MERIN,** Assistant Professor (Sr.Grade),Department of Computer Science and Engineering for her professional guidance and continued assistance during our project.

We express our sincere thanks to the Project Review Committee members, from the Department of Computer Science and Engineering, **Dr. S. REVATHI,** Professor**, Mrs. X. ARPUTHA RATHINA,** AssociateProfessor and **Dr. L. ARUN RAJ,** Associate Professor for their valuable suggestions and support.

We thank our class advisor, **Dr. L. ARUN RAJ,** Associate Professor, Department of Computer Science and Engineering for her guidance and encouragement throughout the project period.

We thank all the **Faculty members** and the **System Staff** of Department of Computer Science and Engineering for their valuable support and assistance at various stages of project development.

**KUSMA THUMMAGUNTA**

**SHAMA HINA.D**

**ABSTRACT**

Data migration using xml as a conversion tool is a system that accepts user input for migrating tables in one database to a different database. Various Databases like MySQL Server, PostgreSQL, MS Access, SQLite are considered for the project. The Databases can also be Oracle or MS SQL Server etc. but not considered as the process would be similar for all of these Databases. All these databases are in various encoding systems and can't be directly migrated. Therefore XML is used for migration of these tables from one database to another database as XML is platform independent and supports all encoding schemes. XML is designed to structure the data and carry it in a rational and sensible manner. This helps web developers and programmers to manipulate the data effectively in a easy way. But, XML can’t be the solution by itself. XML is just a way which is very convenient way to structure and carry data. During the conversion of the data in the tables, the database tables are not changed and stay unaffected. The fields which are retrieved from the tables are next processed for converting into XML. Migration of databases using XML is a project which helps in converting the native Database like My SQL and MS-Access into XML which is a text formatted flat file. This gives the ability to store different kinds of application system data. Using XML for the backend storage helps in reducing the occupied memory space and hence it is not required to pay any additional amount in creating and maintaining the XML database because it is just a file and so requires very less memory space.

Existing Systems of Migration are difficult or complicated processes as that requires the conversion of tables from one database into another database by exporting them as excel sheets or csv files. The disadvantages of it are: it's a Slow and tedious migration process, requires a great deal of expenditure and it's not very user-friendly. The proposed system is considered to be fast migration tool to be implemented using JSP and XML. This system migrates any RDBMS into another RDBMS with an easy GUI. Advantages of it are: it's a really fast migration process, it's completely free as all the technologies are open-source and its very user friendly.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** | v |
|  | **LIST OF FIGURES** | viii |
| **1.** | **INTRODUCTION** | 1 |
| 1.1 | MOTIVATION | 2 |
| 1.2 | PROBLEM DEFINITION | 2 |
| 1.3 | LIMITATION | 3 |
| **2.** | **LITERATURE SURVEY** | 4 |
| **3.** | **METHODOLOGIES** | **10** |
| 3.1 | EXISTING SYSTEM | 10 |
| 3.2 | PROPOSED SYSTEMS | 10 |
| 3.3 | REQUIREMENT ANALYSIS | 12 |
| **4.** | **SYSTEM DESIGN** | **13** |
| 4.1 | SYSTEM ARCHITECTURE | 13 |
| 4.2 | MODULES | 15 |
| **5.** | **LANGUAGES AND TECHNOLOGIES** | 17 |
| 5.1 | CODE AND LANGUAGES USED | 17 |
| 5.2 | UTF-8 ENCODING | 18 |
| 5.3 | DATABASES | 19 |
| **6.** | **IMPLEMENTATION** | **22** |
| 6.1 | DATA EXCHANGE PROBLEM | 22 |
| 6.2 | COMPLEXITIES AND CHALLENGES | 22 |
| 6.3 | ALGORITHM | 23 |
| **7.** | **TESTING** | **28** |
| 7.1 | TEST CASES | 28 |
| 8. | **CONCLUSION AND FUTURE WORK** | 31 |
| 8.1 | CONCLUSION | 31 |
| 8.2 | FUTURE SCOPE | 31 |
|  | **REFERENCES** | 32 |
|  | **APPENDICES** | 34 |
|  | **APPENDIX-A (SOURCE CODE)** | 34 |
|  | **APPENDIX-B (SCREENSHOTS)** | 43 |
|  | **TECHNICAL BIOGRAPHY** | 48 |

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| FIG 4.1.1 | SOFTWARE ARCHITECTURE | 13 |
| FIG 4.1.2 | TECHNICAL ARCHITECTURE | 14 |
| FIG 5.1.4 | IMPLEMENTATION OF JQUERY | 18 |
| FIG 6.3.1 | ACCESSING THE SERVER | 25 |
| FIG 6.3.2 | TABLE RETRIEVAL | 25 |
| FIG 6.3.3 | GENERATION OF XML CODE | 26 |
| FIG 6.3.4 | PREVIEW OF XML FILE | 27 |
| FIG B1. | ADMIN HOME PAGE | 43 |
| FIG B2. | REGISTRATION PAGE | 43 |
| FIG B3. | CHANAGE PASSWORD PAGE | 44 |
| FIG B4. | MIGRATION HOME PAGE | 44 |
| FIG B5. | DATABASE SELECTION BEFORE MIGRATION | 45 |
| FIG B6. | MIGRATION SUCCESSFUL FROM MYSQL TO POSTGRE SQL | 45 |
| FIG B7. | MIGRATION SUCCESSFUL FROM MYSQL TO MYSQL | 46 |
| FIG B8. | POSTGRE SQL PAGE AFTER MIGRATION | 46 |
| FIG B9. | MY SQL PAGE AFTER MIGRATION | 47 |
| FIG B10. | MS ACCESS PAGE AFTER MIGRATION | 47 |

**Chapter 1**

**INTRODUCTION**

Database Migration through XML is a project through which the prevailing databases are migrated to another destination database using XML as a tool. By converting these databases into the XML file format, the databases like Oracle and MS Access for the generation of the XML code are connected inside the network.

Example: Let us consider, Database Administrator holds a firm with multiple companies with each of them having different database. When the database Administrator desires to get all different databasesof thatfirm together in to one single database then the database administrator has to migrateall thesedifferent databases in to one single database.Migrating onedatabase into a different database may be a complicated and hectic hugeprocess. For situations like this our project can often be used, asthe entire database is converted into a XML file first and then into to the desired destination database. Doing this makes the conversioneasier and therefore the source database stays unchanged and protected.To access these databases is allowed by only aauthorized persons.In order to choose the different kinds of databases to be migrated, the database administrator is initially prompted for username and password and only after the authentication, the administrator is allowed to access them. The database administrator will be asked for the database name and then to select all the desired tables present in the database. The database isdisplayed to the administrator after selection, from which he/she can select the table which is wanted to be converted.

Before all this conversion begins, the database Administrator can check if the proper tables are being selected by viewing the tables in the selected database and can make desired modifications. The databases are then converted and the database administrator is received with a message saying migration is complete. A new file is then created and the converted file is going to be in the form of XML format in that file and can be previewed to the database administrator in a browser and then viewed completely inthe database format once migration is complete. During the conversion of the table data, the database containing the tables stays unaffected with out any changes and therefore the fields which are retrieved from the tablesare subjected for the XML conversion. The document overview is a migration technique for converting various databases of various storage formats like Oracle, My SQL, MS SQL Server etc. to the single XML Format as XML is convenient and faster cross-platform application format.

1. **MOTIVATION**

The motivation of database migration project is to analyze the probability of Database migration from different databases into XML Format and to explore the possibility of using XML files as database tables inside the web page. By making use of this project we will convert any database format into one single format called XML, because it is a cross platform, machine independent file format. When the database formats like .mdb,.sql,. dmp, .dbf etc. Cannot interact with one another directly, xml comes for rescue. Moreover, it can easily communicate with front-end languages like C#.NET, VB.NET or Java etc.., by serving as a database for these coding languages. It might be secure but may include constraints also. In our project, we used XML as a database after converting various database table formats from Oracle, My-SQL Server, Derby DB, MS-Access to the single XML Format.

1. **PROBLEM DEFINITION**

Database Migration through XML is first by converting the sourcedatabases like Oracle and MS-Access into the XML database and then to the destination databases like MySQL and PostgreSQL. This project has a capability to store any sort of data from the appliance system since the storage end is a XML file which is in text format. There is great reduction in the storage space and very less memory is occupies since XML is used as a backend. And it doesn’t incur additional costs as XML database is in file format and is easy to maintain with high security using encryption if required. Comparatively, over the other databases XML databases hasmore advantages and comfort.

1. **LIMITATION**

Various databases like MySQL Server, PostgreSQL, MS Access, SQLite are considered in the project. The paid databases like Oracle or MS SQL Server etc. but not considered but the process would be similar for all these Databases.

**Chapter 2**

**LITERATURE SURVEY**

**S. Ramzan et al.,**

This IEEE publication aims at providing an efficient method to transfer the relational database to a big data database and also in improving the data quality. This proposal approach has two phases. The first module is data transformation followed by data cleansing module. This paper provides a solution to overcome the limitations in the traditional databases like unlimited scalability, high performance, data modeling, data distribution, and continuous availability and helps in solving the challenges faced by the business organizations in transforming the conventional databases to NOSQL databases. The future scope of the project is to extend this approach to allow multiple data inputs.

**Winda Sekar Dewi et al.,**

This paper is an international journal which focuses on eliminating the data lost issues through Database migration. Two important techniquesare discussed in the paper to achieve the database migration: a) Data Synchronization and b) Transactional Replication. Data Synchronization helps the migrated database to be synchronized with the original database while Transactional Replication is used to migrate one cloud database to another cloud storage by working on the service area. The evaluation method used in the paper is presented by the comparison of two techniques. The factors influencing the results of migration varies from no.of rows to table size as well as location of the server and also the speed of data uploading and downloading.

**MohdKamirYusof et al.,**

The paper aiming at big data analytics, describes three methods to manage and analyze large amount of structured and unstructured data transfer.  The three approaches used for the data transfer are XML, JSON and through FLAT FILE. The core of the paper is to evaluate the three approaches and find the suitable one to handle large amount of publication data. This paper highlights that the text file format is the best flexible approach huge amount of data while XML and JSON works efficiently and more scalable than FLAT FILE approach. It proves that FLAT FILE approach is the best option for CPU usage as well as Query retrieving speed .

**Alae El Alami et al.,**

This paper explains the conversion of RDB to XML document based on Meta data and semantic enrichment. The RDB is flattened and enriched by using object concepts. This usage of object concept in XML uses the syntax which allows the verification of XML document conformity while creating it. The extracted RDB information is then analyzed, filtered to adjust with the structure of the XML files and the object model associated withit. The implementation in the XML document is built dynamically using SQL Query. A prototype is developed to implement this automatic migration to prove the approach’s effectiveness.

**Kanagaraj.S et al.,**

This paper presents a model to convert Relational Data base to XML database. Through extended entity relationships model, the Relational database schema is converted into XML. EER model helps to map the captured Relational Database schema to XML schema. Various databases like MS - ACCESS, MS- SQL is converted  to XML file format and the file is given to the end user. This paper doesn’t include the semantic constraints in the database and is limited to MS- ACCESS and MS- SQL databases only. The future scope is given to extend this to other databases.

**V. Rajeswari et al.,**

This paper illustrates the development of new technique to store and map XML and different databases by an interactive mapping technique. It is based on XML tested in different scenarios and its verified adaptability in simplifying and integrating data exchange between various database systems. This paper identifies the problems that arise during the creation of integrated interface schema over existing local heterogeneous databases. A new method is proposed to create integrated schema from local database schema. The processing time is calculated by the parser parses the files along with the processing speed calculated on different database types.

**B. Nagarajan et al.,**

This paper is a study on cloud computing in virtual perspective. There are many “on-demand” service models in the cloud out of which cloud storage model lacks a clear structure and neither have a defined set of capabilities nor single architecture. This paper emphasizes it’s study on the storage virtualization taxonomy which is the basis of cloud storage. This paper provides in-depth knowledge in understanding the cloud computing and its service models by reviewing the key technologies like CDMI and OCCI in virtual perspective. This helps in building a compendium in understanding the concept of cloud storage.

**M. Anshar et al.,**

This paper discusses the systematic review on Data migration in the cloud. Data migration has to transfer data secure way and maintain confidentiality in a way there is no data loss due to active attacks. This paper gives a overview of the problems and solutions as well give a proposed solution for migration of data through attunity and helps in optimizing data to replicate and transfer and hence resulting in safer and faster path to accelerate data.

**D. J. Abadi (2009)**

This paper talks about the deploying data management issues on cloud computing platforms and the limitations and opportunities is speculated that large scale data analysis tasks, decision support systems, and application specific data marts are more likely to take advantage of cloud computing platforms rather than transactional or operational database systems. This paper presents a set of features that a DBMS designed for large scale data analysis tasks. This paper also discusses some currently available commercial and open source databases that can be used to perform such analysis, and conclude that none of these presently architecture options match the requisite features. And finally concludes the need for new database management system to be specifically designed for cloud computing environments.

**Ken Ka-Yin Lee et al., (2013)**

This paper aims at providing the three database approaches like NoSQL, native XML and XML- enabled to evaluate the suitability of structured data. Clinical data is highly dynamic, unstructured and often hierarchically and stored in free text. Effective management of clinical data and to transform the unstructured to structured format undergoes various challenging issues. A report is made on database query performance together with the database development experience. The results prove that XML databases are advantageous in terms of scalability, flexibility and extensibility whereas NoSQL is the best option for query speed.

**Marco Mesitiet al.,(2009)**

In this paper, a survey is conducted on  the most interesting and novel approaches for the representation, integration and management of different kinds of biological data by exploiting XML and the related recommendations and approaches. The paper present new and interesting cutting edge approaches for the appropriate management of heterogeneous biological data represented through XML. It concludes saying that XML has succeeded in the integration of heterogeneous bio molecular information, and well acted  as the syntactic glue for biological data sources. Nevertheless, a large variety of XML-based data formats have been proposed, thus resulting in a difficult effective integration of bio informatics data schemes. The adoption of a few semantic-rich standard formats is urgent to achieve a seamless integration of the current biological resources.

**Andrew Clarke et al.,(2012)**

This paper details the use of temporary time stamps and variable hash granularity to increase the efficiency of query assurance. This approach is implemented against data sets of varying type and size, including encrypted data to illustrate the potential overhead issues present in distributed systems and data repositories. This paper presents  empirical study detailing XML as suitable approach for emerging information system architectures, especially for health information systems. This work explored the area of secure and efficient query assurance in external and distributed XML databases. Procedures to provide that assurance while reducing data overheads compared to similar approaches were investigated.

**N. Li et al., (2011)**

This paper focuses on the heterogeneous data exchange problems in the Internet of Things Sharing Platform such as the interconnection of different database systems problem and also provide the data sharing and data transparency services. This paper proposes a data conversion technology based on mapping of relationship schema. This helps in analyzing the problems faced in the data sharing platform and studies the practical technology of XML heterogeneous data exchange. A specific implementation mechanism is developed to solve problems like relationship misjudging, message losing and field attribute change during data exchange.

**R. Ahmadi et al., (2012)**

This paper addresses one of the major challenges in database structure development i.e. the process of data migration. The crucial problems are huge database size and syntax and semantic differences in the tables. The two crucial modules of data migration are structure mapping and migration in a sequence. Traditionally the first phase is carried out by humans and the second by the software whereas in the new approach, in the first phase, the process is carried out by ontology automatically. This paper provides the study of various methods and a new model for migrating with different structure is proposed. In this new model, a middle layer is created by covering both traditional and semantic databases and this is the highlight of the proposed system.

**Chapter 3**

**METHODOLOGIES**

1. **EXISTING SYSTEM**

The Existing system is very helpful only for single database conversions. Databases are most commonly in wide usage in different resource types in these days. A part of the server memory is hence occupied by these databases and this leads to additional costs in order to maintain these databases. In web, databases are most widely in usage at the present times. The amount is based upon the factors like size of the database, and also Oracle and Access databases are platform dependent.

1. **Disadvantages in Existing system**

This process is complex and has subsequent drawbacks :

* Data corruption will occur.
* Requires huge memory space.
* Incurs additional expenses for database security.

Inside the server, a vast ammount of memory space is occupied by these databases and hence some amount of cash has to be paid to the service provider to keep up that database. The money to be paid differs accordingly depending upon the size of the databaseand also throws many challenges as different problems arise during the maintenance the database. There is a constantneed to make database replication and database backup has to be performed constantly from time to time and it accounts to hugememory space.

1. **PROPOSED SYSTEM**

Database Migration through XML is migrating the prevailing database like Oracle and MS-Access into new desired destination database with the help of XML as a conversion tool. There is great reduction in the storage space and very less memory is occupies since XML is used as a backend.

The XML has the power to be processed on almost any platform because it is platform independent and hence we are able to use this in different kinds of operating system. The data in XML format can’t be restricted by any proxy server or Firewall. Here maintenance is extremely easy asthere is very less memory space is occupied as it is in file format.

Earlierthe maintenance of the database was a hectic task and also huge added costs. So many databases aren't supported in several platforms but XML file format supports different platforms.

1. **Advantages of Proposed system**

The advantages of proposed system are:

* Occupies very less memory space.
* No additional costs for maintaining the database.
* Since this proposed system modifies the contents of the database into a text file, there isn’t much need of administrator support.
* There is not much data loss since the database is converted into plain text format.
* This is very much compatible in any OS as it is cross platform independent.
* Human resource requirement is barely minimum.
* The accessing speed is going to be fast during the processing of data.
* Though it’s a flat file when compared to database storage it provides enhanced security. Enhanced security is assured since we offer Authentication while retrieving tables from the Databases.

1. **REQUIREMENT SPECIFICATION**

Requirement analysis and Specification has a vital importance in creating a good quality software solution for the problem encountered. To assess the clarity of the software the requirements are to br refined and analyzed very carefully. For the implementation of the successful software, it is very much needed for the requirements to be represented in very clear way. Each and every mentioned requirement should be very consistent with the overall stated objective. This project development consists of the following requirements.

* Software Requirements
* Hardware Requirements

1. **Software Requirements**

|  |  |
| --- | --- |
| Operating System | : Windows 10 |
| Coding Language | : JAVA 1.7, JSP Servlets |
| Frontend | : HTML5, JQuery, JavaScript |
| Webserver | : Apache Tomcat 7.0 |
| Database | : MySQL Server 5.1, MS Access, SQLite, PostgreSQL |

1. **Hardware Requirements (Minimum)**

|  |  |
| --- | --- |
| Processor | : Pentium-IV 2.4GHz or above |
| RAM | : 1 GB |
| Hard Disk Drive | : 160GB |

**Chapter 4**

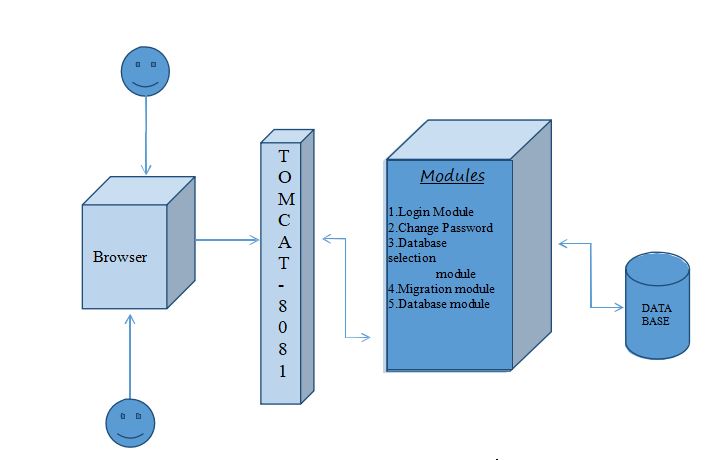
**SYSTEM DESIGN**

1. **SYSTEM ARCHITECTURE**

During System design, there is a the program structure, data structure and procedural details needed to be refined progressively and developed. They should then be reviewed before documenting them. There are two perspectives from which a system design can be viewed. They are technical perspective or project management perspective.

1. **Software Architecture**

Software architecture defines the high level structures of the software and then helps in constructing the process of those structure creation and then the documentation. These structures are essential to explain about the software. There are software elements in each structure. It is used to define the relations among those elements and to explain their properties. It also helps in describing the connectivity between elements and relations. The software architecture is shown in Figure 4.1.1



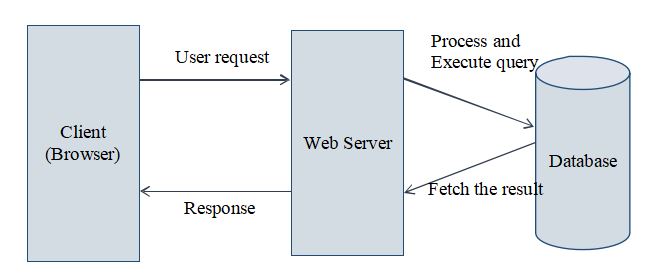
**FIG 4.1.1SOFTWARE ARCHITECTURE**

This Software architecture and its documentation facilitates communication between stakeholders, and also helps in capturing the early decisions regarding the high level design. This facilitates the reusage of these components design in another projects.

1. **Technical Architecture**

Technical Architecture of an application helps to design and document the software application. When developers want to build the system from the scratch or modify the existing computer system, this architecture can be used to create a schematic blueprint which helps in achieving the target very easily.

The layer of the system architecture of the computer which helps in defining and specifying the protocols, interfaces and parameters used in the system and product architecture layers are included in this technical architecture. Before building a software application, it is important that the computer architecture to be defined. The communication network used by the application is typically defined using the technical architecture. The technical architecture is given in the Figure 4.1.2



**FIG 4.1.2TECHNICALARCHITECTURE**

1. **MODULES**
2. **Database Selection Module**

This module is responsible for selecting a particular source database and destination database for migration. There are a set of databases listed under “select source databases” drop down button and similarly a set of destination database under “select destination button”. After choosing the destination database, the migration starts and enters the next module. For additional confirmation, a pop up can be added with the dialogue with source and destination databases.

1. **Migration Module**

This module is responsible for migration of tables from a source database to a destination database. This module is initiated once the user confirms the source and destination databases. A request is sent to the server for the auto execution of migration JSP code process. This module converts the entire data of source Db into XML format and then inserts XML format generated by source DB into destination DB in its format.

1. **Database Module**

This module is responsible for storing the tables in various databases after or before migration. This module handles the key constraints, schema specifications and storing the migrated information. This module helps in solving the duplicate table problems if migration occurs more than once.

1. **Login Module**

This is an authentication module for entering into the migration software. The login details are also stored in the xml format and no backend database is used to store these login details.

1. **Change Password**

This is for providing better security for the admin password, so that the password is changed regularly for better authentication. This module is restricted for other users as the information in the databases may be confidential and can be misused and hence only admin can register new people and change their passwords and give permissions to another registered in case the admin wants to quit.

**Chapter 5**

**LANGUAGES AND TECHNOLOGIES**

1. **CODE AND LANGUAGES USED**
2. **Java**

This gives a brief of what and why Java language is used in this project. It is intended for the developers where the code which runs on one platform need not be compiled again to run on the other. Java applications can typically run on any Java virtual machine despite the architecture of the computer by compiling to byte code. Especially for client- server web applications, java is one of the most popular programming languages in use. Here, for the application it is needed to retrieve the required tables and data from My SQL DB which is the source database.

1. **HTML**

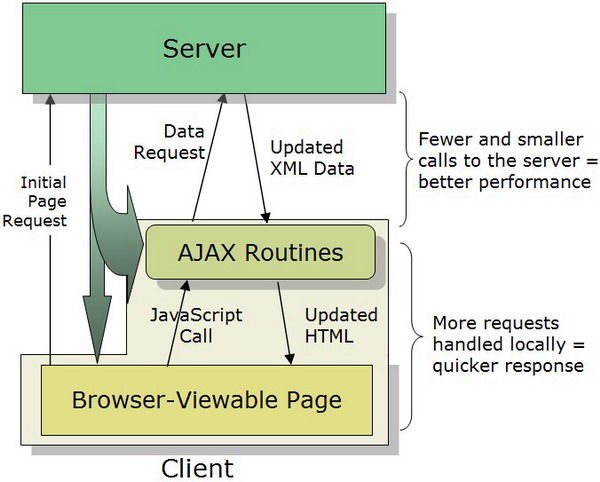
The main purpose of using Hypertext Mark-up Language (HTML), allow users in producing the web pages which includes text, pointer and graphics which redirect to other webpages using hyperlinks. The main point of using hyperlinks is to link one point into the another one.

1. **Apache**

Apache and MySQL modules are started as server application and database. Web server acts as a proper tool as middleman between clients and server communication .

1. **JQuery**

Using JavaScript on websites is made easy by JQuery. It is a set of JavaScript methods and functions. It is light weighted and helps in wrapping many JavaScript lines into methods which can be called by single line code. It simplifies many complex JavaScript’s, CSS manipulations, AJAX calls and HTML/DOM manipulations. JQuery has many plugins for all the complex tasks.



**FIG 5.1.4 IMPLEMENTATION OF JQUERY**

1. **Ajax**

AJAX is the used for server data exchange and to update webpage modules without reloading the whole page. It loads data in the background to display it on the webpage. It doesn’t require to reload the entire page again. JQuery provide various methods for AJAX functionality. Together with JQuery, AJAX is used in requesting text, HTML, XML from remote web servers using HTTP Get or Post to in loading the external data directly to the web page HTML elements.

1. **UTF-8 Encoding**

**Unicode Text Format – 8 :**

Unicode format is a standard in computing industry for representing and to handle the text which is expressed in most of the writing systems in the world and for consistent encoding. The latest version of Unicode contains the repertory of over 110,000 characters which covers a range of 100 scripts and many other symbols are being developed in the conjunction with the Universal Character Set standard. There is an encoding method in each standard with set of reference data computer files, a standard encoding set and for the reference of vision it includes a set of charts with code. It also contains a huge number of items related to the standards like rules for normalization, properties of the characters, collation,rendering,decomposition, a display order which is bidirectional. The success of the Unicode’s character sets which are unifying is the main reason why the computer software localization and in internationalization. In many recent technologies, including many modern operating systems XML, Java, and the Microsoft has implemented this standard.

By using various types of character encoding, Unicode can be implemented. UTF-8, UTF-16 are the most widely and predominently used encodings and the now-obsolete UCS-2. For any ASCII character, one byte is used by UTF-8 where all have the same values for the code either in ASCII or UTF-8 and for the other characters, it takes up to four bytes. For each character 16-bit code unit (two 8-bit bytes) is used by the UCS-2 but in the current Unicode standard, not every character can be encoded. UCS- 2 is extended by UTF-16 , for the characters which can be represented in UCS-2, one 16-bit unit is used and to handle each of the other characters which are additional, two 16-bit units are used to handle them.

1. **DATABASES**
2. **My SQLServer**

MySQL server is a database management system which acts like a client and server system. My SQL server provides capabilities of querying and connectivity, along with the ability of excellent structuring of data and for many different platforms integration. This MySQL server is very suitable for accessing databases as it provides very rich functionalities like the speed, connectivity and also the security. For highly demanding environments of production, it helps a lot in handling very large databases in a quick and reliable manner. For supporting different administrative tools, varied back ends, client programs and libraries, and also many other API’s there is a SQL server in the system which is multi-threaded.

1. **SQLite**

The programming library of C contains the SQLite. Unlike any other database management systems, SQLite doesn’t work like a client–server database engine. Rather, it is embedded into the end program.

There are no standalone processes with which the application program communicates in the SQLite engine unlike any other client–server database management systems. The SQLite library which can also be called dynamically, instead is linked in and turns out to be application program’s integral part. Through simple function calls, the SQLite’s functionality is used by the application program, which helps in reducing database access latency. There is more efficiency in a single process function calls rather than that of in inter-process communication. On the host machine, the entire database (definitions, tables, indices, and the data itself) is stored in the SQLite as a single cross-platform file. Though SQLite writes can be only sequentially performed read operations can be multitasked. By locking the entire database during the write operation, this simple design is implemented.

1. **PostGRE SQL server**

PostgreSQL or simply Postgres is an open-source software and an object-relational database management system (ORDBMS) with focuses on standards- compliance and also extensibility. Many workloads with the range starting from very small single-machine applications to very large Internet-facing applications which consists of many concurrent users can be handled. Working as a database server, the main functionality is to store data, securely also for supporting the best practices,It also allows the retrieval on the request of other software applications. Being a cross-platform, PostgreSQL can run on many different operating systems including Linux, FreeBSD ,Solaris , and Microsoft Windows. The vast majority of Linux distributions have it available in supplied packages.

All the databases which are newly created will by default have a schema named “public”. Also the public schema isn’t mandatory and any additional new schemas can be added. The system check order for the unqualified objects schemas (those without a prefixed schema) is determined by a "search path"; Search paths can be configured on a database or role level. The search path by default contains the special schema “$user” which first looks for schema named after the user in the connected database. If such schema isn't found, then the next schema in the list is proceeded to. Which ever valid schema is found first in the search path, new objects are created in it.

**Chapter 6**

**IMPLEMENTATION**

1. **DATA EXCHANGE PROBLEM**

This algorithm is constructed in a way to solve data exchange problem. Migrating the database contents to the target schema instance that reflects the data of the source base as accurately as possible. This problem of data exchange is several decades old, and now once again become very important when XML has been created , a data format which was created to solve the data exchange problem.

The basic problem here is to recognize two things : a) Which data from the source should be transferred and b) To where in the target must the source data go into. In the sense, with two major differences this partly has the problem of schema integration. Firstly, There is no need to create a new schema, the target schema is already present. Secondly, on contrary to the case of the schema integration problem, there persists a set of own constraints on the target schema that has be dealt with during this data migration.

1. **COMPLEXITIES AND CHALLENGES**

There are different databases types that are actually used in the Real-time. When there is the need for interoperation between databases, that’s where the main complexity arises. The entire database or some part it has be migrated to a common database at this stage. There will be a overhead for sure if the databases are too large. Some part of the memory is occupied by these databases. So in order to maintain the databases ,it requires the users to pay some amount to the site owner.

The data exchange problem is because of the two main reasons. Firstly, in contrast with the XML files, there is very less structural data in relational database, which means that in many cases, in the XML files, the structural data is huge than the actual data. To find the associations between data items, this huge amount of structural information can be exploited. Secondly, by using namespaces, RDF and ontologies, data migration in the future might be fast, error-free and effortless.

1. **ALGORITHM:**

Login

* Select source and destination databases

Auto Execution of Migration JSP code process

* converts the entire data of source Db into XML format
* Inserts XML format generated by source DB into destination DB in its format

Generation of XML Format from SourceDBProcess:

* Connect to the jdbc driver
* If table=sourceDB,get column names, constraints and data

Create\_tags\_xml process:

* Insert table name as the root tag in xml
* Get the table details to create xml tags
* Get the schema of table to insert into xml

Insert XML format data into DestDB process:

* Connect to jdbc driver
* Parse the xml file of source database
* Get the table name from the root tag
* All the column names are obtained by considering the child tags
* Insert the values into table

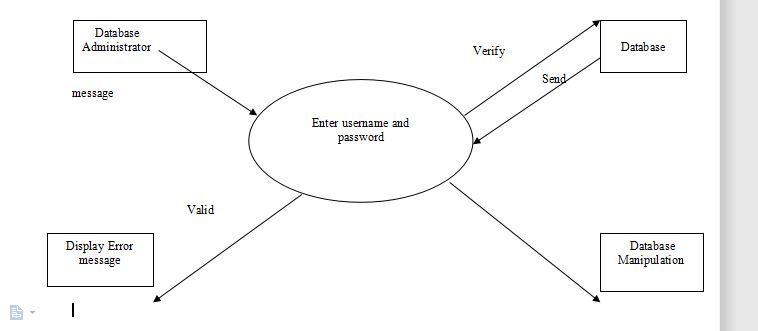
There are two assumptions made in the implementation of the proposed method. Firstly, each element of the XML file must be related semantically to both of its parent and as well as the children, hence related to its ancestors and descendants by transitivity. Secondly, in an XML file, since any two elements will share a common ancestor and so any two elements are in fact semantically related, even if one of them doesn’t fall in the ancestors-descendants path of the other one.

The algorithm inputs is the XML source file, the data source schema of the XML, the target data source schema of the XML, and also includes file which specifies the mappings between the names of the source and target data source schema elements. Hence, to migrate this data, at each step the algorithm has to be constructed with the knowledge of the elements it will have to import from the source file so as to finally insert that data into the target file. In order to do that, the algorithm has to traverse the target Schema and for every node that will be encountered, create an appropriate XPath expression and fetch the appropriate node from the source. The enhanced database migration process has the following steps:

* Accessing the Server
* Retrieval of tables
* XML Code generation
* Preview of XML file

1. **Accessing the Server:**

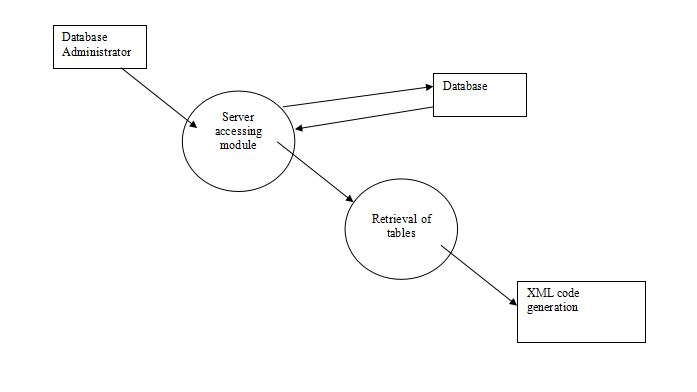
SQL server consists of its ownusername (root access) and password, for accessing the database file. It has high security which means with improper user name and password, the database files in SQL server can’t be accessed. During the conversion of databases into XML file format, the database like MS-Access gets connected in the network which then generates the XML code. Authentication must be provided for user the and also should also protect by restricting the unauthorized persons accessing the server database. MS-Access database can also be connected through the local drivers, floppy or compact disk. The function of accessing the server is shown in Figure 6.3.1.



**FIGURE 6.3.1 ACCESSING THE SERVER**

1. **Tables Retrievals:**

The structure of the table can be seen by selecting the database table each. In particular, the displayed structural data that are data type, size, fieldname, scale and precision. By collecting these are used to verify the cascading style sheet(CSS). If any need to modify the database is identified then select the table, manipulate it using the operations like insert, update and delete.

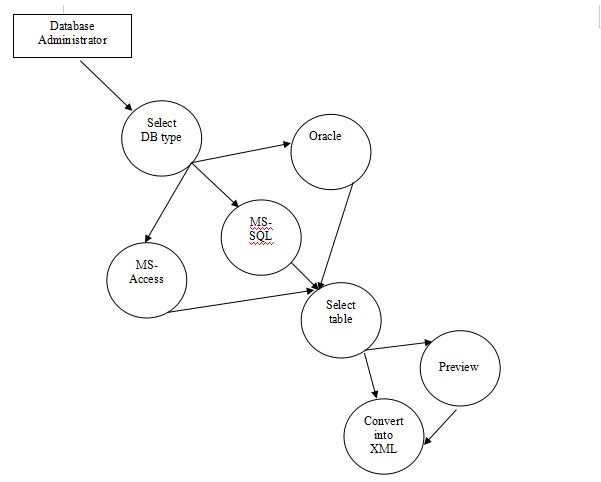


**FIGURE 6.3.2 TABLE RETRIEVAL**

After the user authentication, selection of the database type is allowed by the user and hence prompted for choosing the database name. The User also has the choice of selecting the table which needs to be converted and also be preview it to reduce malselections. The process of retrieving the tables is shown in Figure 6.3.2.

1. **Generation of XML code:**

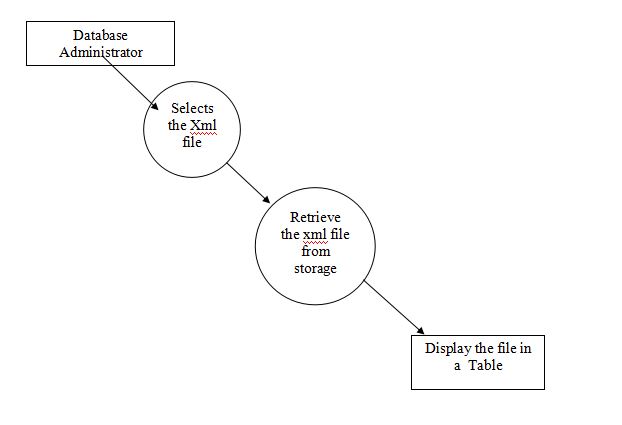
The table data is first selected and then is proceeded for the conversion into XML file format. For the process of conversion, the selected table has to be assigned implicitly to a temporary file. By using the appropriate code, that file which is assigned is then converted into a XML file. Even after the conversion, the original database’s data will remain unchanged. The tables are retrieved from the database, for each and every field individual tags are created as just by the user specifications. This step is used to generate the XML code for the database and it is shown in Figure 6.3.3



**FIGURE 6.3.3 GENERATION OF XML CODE**

1. **Preview of XML file:**

To view in XML format, the XML file name is prompted to the user and by doing this it can check if all the respective table fields are converted as requested by the client. If any error occurs, the XML file which is generated can be modified and the of XML file preview is shown in Figure 6.3.4 and Still therewont be any change in the source file.



**FIGURE 6.3.4 PREVIEW OF XML FILE**

**Chapter 7**

**TESTING**

Since the errors in the software can cause trouble at any stage, we have to carry out the testing process at different levels during the development. The basic levels of testing are unit, integration, system and acceptance testing. The unit testing is carried out on coding. Here different modules are tested in the table 7.1 aagainst the specifications produced during design of the modules. In case of integration testing, different tested modules are combined into sub systems and tested. In case of system testing, the full software is tested and in the next level of testing, the system is tested with user requirement document prepared during SRS.

1. **TEST CASES :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Test** | **Input** | **Obtained Output** | **Actual Output** | **Description** |
| 1 | Valid Login | User ID,  Password | Login Success | Login Success | Test Passed!  control transferred to Admin Home page. |
| 2 | Invalid Login | User ID,  Password | Login Failed | Login Failed | Test Passed! Try Again |
| 3 | Invalid Login | Null, Null | Login Failed | Login Failed | Test Passed! Try Again |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | Invalid Password, Password updation Failed | Old Password,  New password & Confirm Password | Failed | Failed | Test Passed!  Old password incorrect or new password & confirm password mismatch. |
| 5 | Valid Password, Password updation | Oldpassword  new Password &Confirm Password | Success | Success | Test Passed! Password Changed |
| 6 | Test for proper and complete migration of a database from SQLite to MySQL  server. | Input of SQLite and MySQL server as source and destination databases. | Passed | The application has to completely migrate a specificschema in SQLite into a schema in MySQL Server. | It is properly and completely migrating a specific schema in SQLite into a schema in MySQL Server. |
| 7 | Test for migration of database from MySQL server to PostGRE SQL  Server | Input of  MySQL server and PostGRE SQLServer as source and destination databases. | Passed | The application has to completely migrate a specific schema in MySQL Server into a schema in PostgreSQL Server. | Yes, The application is Properly and completely migrating a specific schema in MySQL Server into a schema in Post GRE SQLServer |
| 8 | Test for migration of a database from MySQL server to MS Access. | Input of MySQL server and MS Access as source and destination databases | Passed | The application has to completely migrate a schema in MYSQLServer into a schema in MS Access | Yes, The application is Properly and completely migrating a specific schema in MySQL Server into a schema in MS Access |
| 9 | Test for preserving constraints like ‘primary key’ etcand the data types of each field. | Check the constraints after migration | Passed | The Application should Preserve constraints like ‘primary key’ etc and the data types of each field. | Yes, The Application is Preserving constraints like ‘primarykey’etcand the data types of eachfield. |
| 10 | Testing the Application for proper migration of each table data in each case of migration. | Comparing the databases before and after migration. | Passed | The application should properly migrate each table data in each case of migration properly. | Yes, The application is properly migrating each table data in each case of migration properly. |

**TABLE 7.1 : TEST CASES**

**Chapter 8**

**CONCLUSION AND FUTURE SCOPE**

1. **CONCLUSION**

By making use of this project we can convert any database format into one single format called XML, as it is a cross-platform, machine independent file format . When the database formats like .mdb, .sql, .dmp, .dbf etc.. Cannot interact with each other directly, xml comes for rescue. Moreover, it can easily communicate with front- end languages like C#.NET,VB.NET or Java etc.., by serving as a database for these coding languages. It could be secure and can include constraints also. In our project, we used XML as a database after converting various database table formats from Oracle, My-SQL Server, Derby DB, MS-Access to one single XML Format. In this project there are only two users , The database Administrator and the .NET programmer.

1. **FUTURE ENHANCEMENT**

The Project could be extended further to include XSD/XS schema related conversions for further enhancement of the XML file format. Using this schema-based XML Format would greatly improve the security of XML File at file level.

REFERENCES

1. ShabanaRamzan,ImranSarwarBajwa et al.,“Intelligent Data Engineering for Migration to NoSQL Based Secure Environments”, [IEEE Access](https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=6287639) Advanced Software and Data Engineering for Secure Societies, Vol.7, pp.69042- 69057, 2019.
2. Winda Sekar Dewi, EmaUtami, BambangSudaryatno,“Database Migration using Data Synchronization and Transactional Replication”, International Journal of Innovative Technology and Exploring Engineering, ISSN: 2278-3075, Volume-8 Issue-10, August 2019.
3. MohdKamirYusof, Mustafa Man, “Efficiency of FlatFile Database Approach in Data Storage and Data Extraction for Big Data”, Indonesian Journal of Electrical Engineering and Computer Science, Vol. 9, No.2, pp.460~473, February 2018.
4. Alae ElAlami, Mohamed Bahaj, “Schema and DataMigration of a Relational Database RDB to the Extensible Markup Language XML”, World Academy of Science, Engineering and Technology ,International Journal of Computer and Information Engineering , Vol:9, No:7, 2015.
5. KanagarajS,DrSunithaAbburu,“ Converting Relational Database Into Xml Document”. IJCS International Journal of Computer Science Issues, Vol. 9, Issue 2, No.1, March 2012.
6. V. Rajeswari, Dr. Dharmisthan, K. Varughese, “A Novel Approach for Integrating Heterogeneous Database through XML.” International Journal of Computer Science and Information Technologies, Vol. 2 (2), 633 – 640, 2011.
7. B. Nagarajan and Dr.J. Suguna, “A Review on Cloud Data Storage in Virtual Perspective” in International Journal of Computer Science and Information Technologies, vol. 5, pp. 6027-6031, May 2014.
8. M. Anshar, M. W. Ashraf, M. Fatima, “Data Migration in Cloud : A Systematic Review” in American Scientific Research Journal For Engineering Technology and Sciences (ASRJETS), vol. 48, no. 1, pp. 73-89, 2018.
9. D. J. Abadi, ‘‘Data management in the cloud: Limitations and opportunities,’’ IEEE Data Eng. Bull., vol. 32, no. 1, pp. 3–12, Mar 2009.
10. Ken Ka-Yin Lee, Wai-Choi Tang, Kup-Sze Choi, “Alternatives to relational database: Comparison of NoSQL and XML approaches for clinical data storage”, Computer Methods and Programs in Biomedicine, Vol 110, pp: 99-109,2013.
11. Marco Mesiti, Ernesto Jiménez-Ruiz, “XML-based approaches for the integration of heterogeneous bio-molecular data”, BMC Bioinformatics , 2009.
12. Andrew Clarke, Eric Pardede, Robert Steele, “External and Distributed Databases: Efficient and Secure XML Query Assurance”, International Journal of Computational Intelligence Systems, [Volume 5, Issue 3](https://www.atlantis-press.com/journals/ijcis/issue/197),Pages 421 - 433, June 2012.
13. N. Li, B. Xu, X. Zhao, and Z. Deng, ‘‘Database conversion based on relationship schema mapping,’’ in Proc. Int. Conf. Internet Technol. Appl. (iTAP), pp. 1–5, Aug 2011.
14. R. Ahmadi, B.R. Cami, H. Hassanpour, “Automatic Data Migration between Two Databases with Different Structure”, in International Journal Applied Information Systems (IJAIS), vol. 3, no. 3, July 2012.
15. M. C. Mazilu, “Database Replication”, in Database Systems Journal, vol. 1, no. 2,pp. 33-38, 2010.

**APPENDIX - A**

**PSEUDO CODE**

#### 1. Login

**// If the user credentials are valid, login is successful.**

if(user1.equals("admin")&&pass1.equals("admin1234"))

{

i=i+1; response.sendRedirect("AdminHomePage.jsp");

}

return false;

}

#### 2. Execution of Migration JSP codeprocess

Using AJAX coding:

xmlHttp=GetXmlHttpObject()

if(xmlHttp==null) //if xmlHttp=null the request is notsupported

{

alert ("Browser does not support HTTP Request") return

}

if(val=="SourceDB") //if val=SourceDB redirect to “sourcedata.jsp" varurl=”sourcedata.jsp"

#### // converts the entire data of source Db into XML format

elseif(val=="DestDB") // if val=DestDB redirect to "destinsert.jsp” varurl="destinsert.jsp"

#### // Inserts XML format generated by sourceDB into destination DB in its format

//url=url+"?results="+valxmlHttp.onreadystatechange=stateChangedxmlHttp.open("GET",url,true) xmlHttp.send(null)

#### 3. Generation of XML Format fromSourceDBProcess

class.forName(“jdbc.driver.sourceDB”);

#### // connect to the jdbc driver

con=DriverManager.getConnection("jdbc:sourceDB://localhost:3307/sourcedbschema","username","pwd");

rs = st.executeQuery("select table\_name from information\_schema.tables where table\_type='base table' and table\_schema='sourceDBSchema'");

#### // If table=sourceDB, get column names,constraints and data

while(rs.next()){

create\_tags\_xml(rs.getString(table\_name)); create\_tags\_xml(rs.getMetaData().getColumns()); create\_tags\_xml(rs.getMetaData().getConstraints()); create\_tags\_xml(rs.getString(cloumn\_data));

}

#### 4. Insert XML\_Format Data into DestDB process

Class.forName("jdbc.Driver.DestDB");

#### // connect to jdbc driver

con=DriverManager.getConnection("jdbc:DestDB://localhost:portno/DestDBSckusma", "username","pwd");

#### // parse the xml file of source database

DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance(); DocumentBuilderdb = dbf.newDocumentBuilder();

Document document = db.parse("webapps\\ROOT\\sourceDBdata.xml"); NodeListnltn= document.getElementsByTagName("Tname"); NodeListnlcn= document.getElementsByTagName("column\_name");

NodeListnldt= document.getElementsByTagName("DType"); NodeListnldl= document.getElementsByTagName("DLength"); NodeListnlpk= document.getElementsByTagName("PKey");

#### // All the column names are obtained by considering the child tags

NodeListchildList = flowList.item(0).getChildNodes(); for (int j = 0; j <childList.getLength(); j++) {

Node childNode = childList.item(j); NodeListrowChild = childNode.getChildNodes();

for (int r = 0; r <rowChild.getLength(); r++) {

Node rNode = rowChild.item(r); tblCols.add("'"+rNode.getFirstChild().getNodeValue()+"'");

}

StringBufferconcol = new StringBuffer(); for(int cl=0;cl<tblCols.size();cl++){ if(cl!=tblCols.size()-1)

concol.append(tblCols.get(cl)+","); else

concol.append(tblCols.get(cl));

}

out.println(concol.toString());

#### // Insert the values into table

st4.executeUpdate("insert into "+tables.get(z)+" values("+concol.toString()+")"); tblCols.clear();

} z++}

**5**.**create\_tags\_xml() process :**

using JDK packages javax.xml.parsers.\*,javax.xml.\*,javax.xml.transformer.\*:

DocumentBuilderFactorydocumentBuilderFactory= DocumentBuilderFactory.newInstance();

DocumentBuilderdocumentBuilder = documentBuilderFactory.newDocumentBuilder(); Document document = documentBuilder.newDocument();**// Insert table name as the root tag in xml**

Element root = document.createElement("Tables");document.appendChild(root);

rows = document.createElement(rs.getString(1));

//rows.appendChild(document.createTextNode());table.appendChild(rows); Element tn = document.createElement("Tname");tn.appendChild(document.createTextNode(rs.getString(1)));rootElement.appendChild(tn);

**// Get the table details to create xml tags**

rs5=st3.executeQuery("select table\_name,column\_name,data\_type,character\_maximum\_length,column\_type,column\_key from information\_schema.columns where table\_name='"+rs.getString(1)+"' and table\_schema='user\_details'");

**// Get the schema of table to insert into xml**

while(rs5.next()){ out.println(rs5.getString(2)+rs5.getString(3)+rs5.getString(4)+rs5.getString(6));Element cname = document.createElement("col\_name"); cname.appendChild(document.createTextNode(rs5.getString(“column\_name”)));tn.appendChild(cname);Element dtype = document.createElement("DType"); dtype.appendChild(document.createTextNode(rs5.getString(“data\_type”))); tn.appendChild(dtype);Element dlen = document.createElement("DLength"); dlen.appendChild(document.createTextNode(rs5.getString(“data\_length”))); tn.appendChild(dlen);if(!rs5.getString(6).equals("PRI")){Element pkey = document.createElement("PKey"); pkey.appendChild(document.createTextNode("")); tn.appendChild(pkey);}

else

{

Element pkey = document.createElement("PKey"); pkey.appendChild(document.createTextNode(rs5.getString(“primary\_key”))); tn.appendChild(pkey);

}

Element name = document.createElement(rsMetaData.getColumnName(i)); name.appendChild(document.createTextNode(rs1.getString(“cloumn\_data”)));row.appendChild(name);

}

**6) Insert XML\_Format Data into DestDB process:**

Class.forName("jdbc.Driver.DestDB");

**// connect to jdbc driver**

con= DriverManager.getConnection("jdbc:DestDB://localhost:portno/DestDBSchema",

"username","pwd");

**// parse the xml file of source database**

DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();DocumentBuilderdb = dbf.newDocumentBuilder();Document document = db.parse("webapps\\ROOT\\sourceDBdata.xml");

NodeListnltn= document.getElementsByTagName("Tname");NodeListnlcn= document.getElementsByTagName("column\_name");NodeListnldt= document.getElementsByTagName("DType");NodeListnldl= document.getElementsByTagName("DLength");NodeListnlpk= document.getElementsByTagName("PKey");

**//Get the table name from the root tag**

Node root= document.getDocumentElement();NodeListnlda= root.getChildNodes();NodeListtrow = document.getElementsByTagName("Row");

**// All the column names are obtained by considering the child tags**

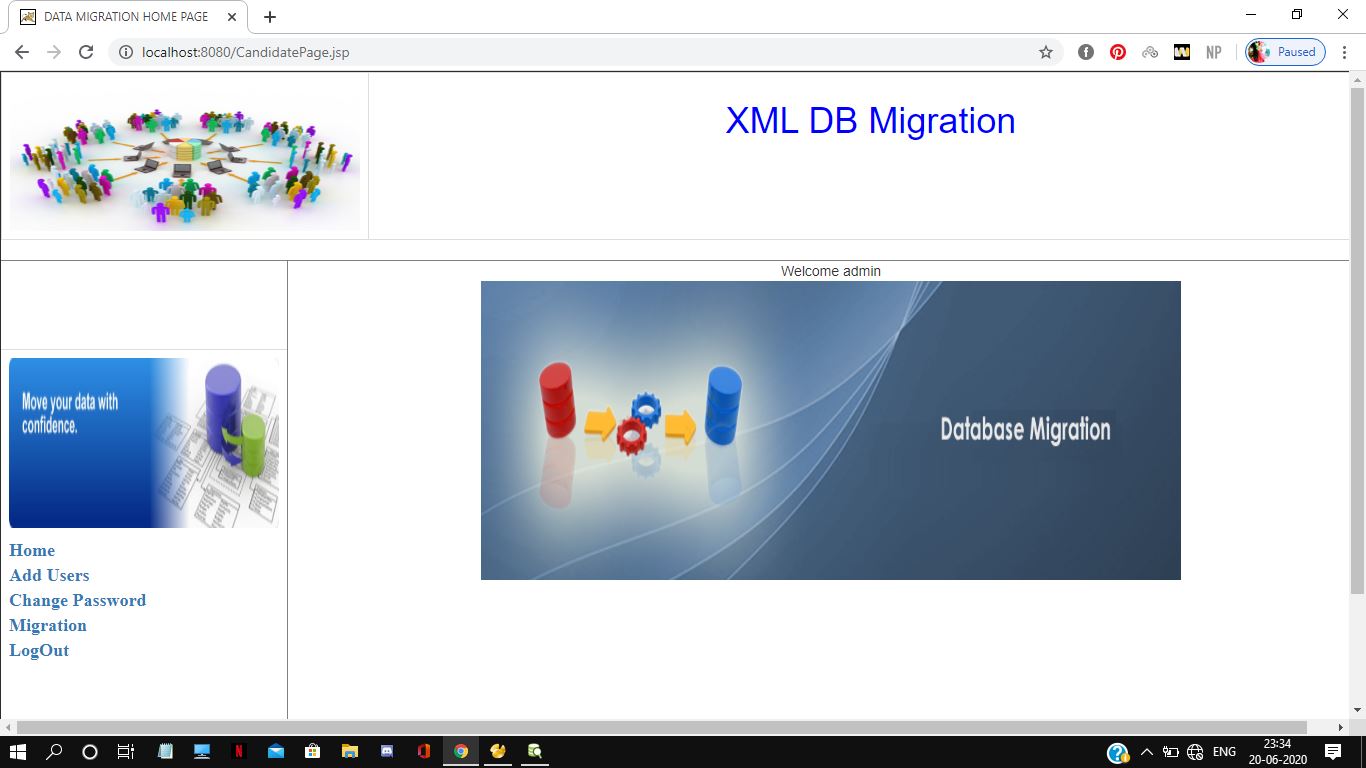
NodeListchildList = flowList.item(0).getChildNodes();for (int j = 0; j <childList.getLength(); j++) { Node childNode = childList.item(j);NodeListrowChild = childNode.getChildNodes(); for (int r = 0; r <rowChild.getLength(); r++) { Node rNode = rowChild.item(r); tblCols.add("'"+rNode.getFirstChild().getNodeValue()+"'"); } StringBufferconcol = new StringBuffer(); for(int cl=0;cl<tblCols.size();cl++){ if(cl!=tblCols.size()-1)

concol.append(tblCols.get(cl)+",");else concol.append(tblCols.get(cl));}out.println(concol.toString());**// Insert the values into table**

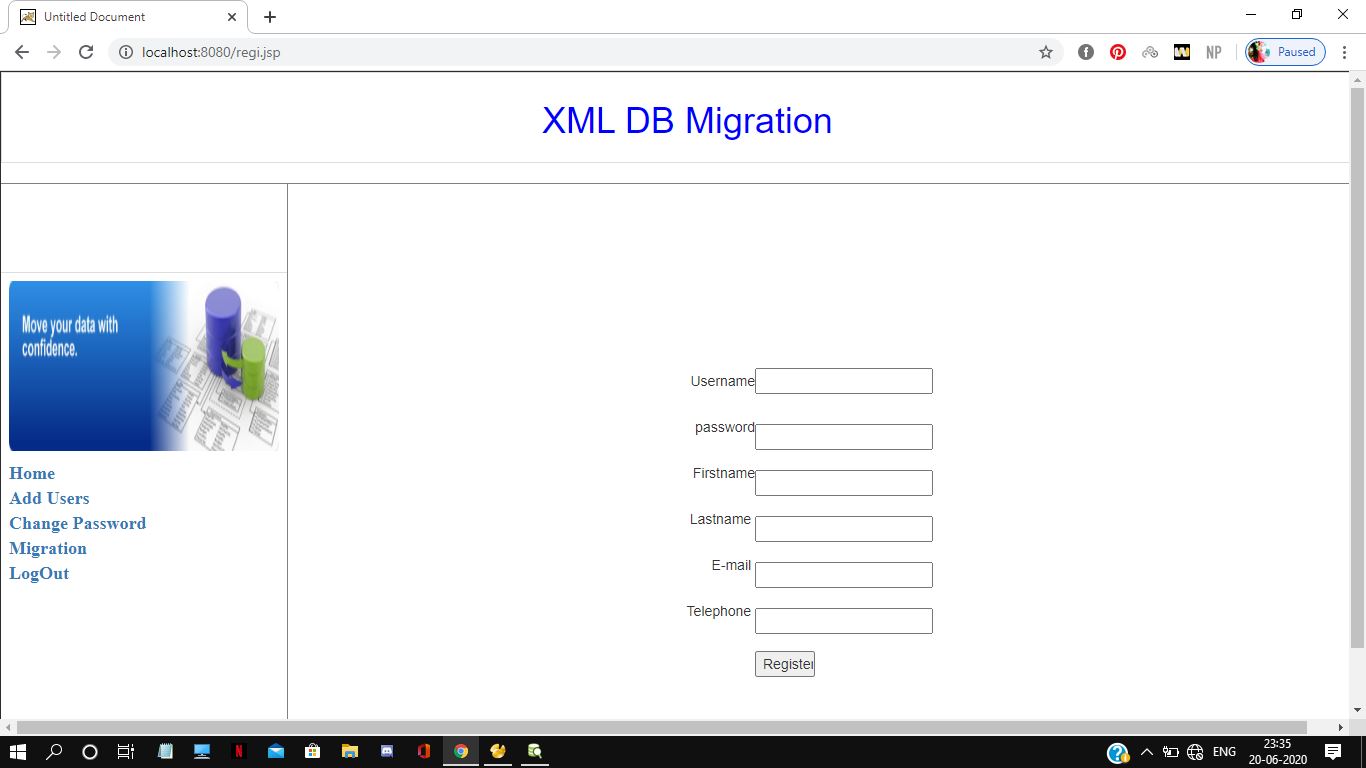
st4.executeUpdate("insert into "+tables.get(z)+" values("+concol.toString()+")"); tblCols.clear();}z++;}

**APPENDIX B**

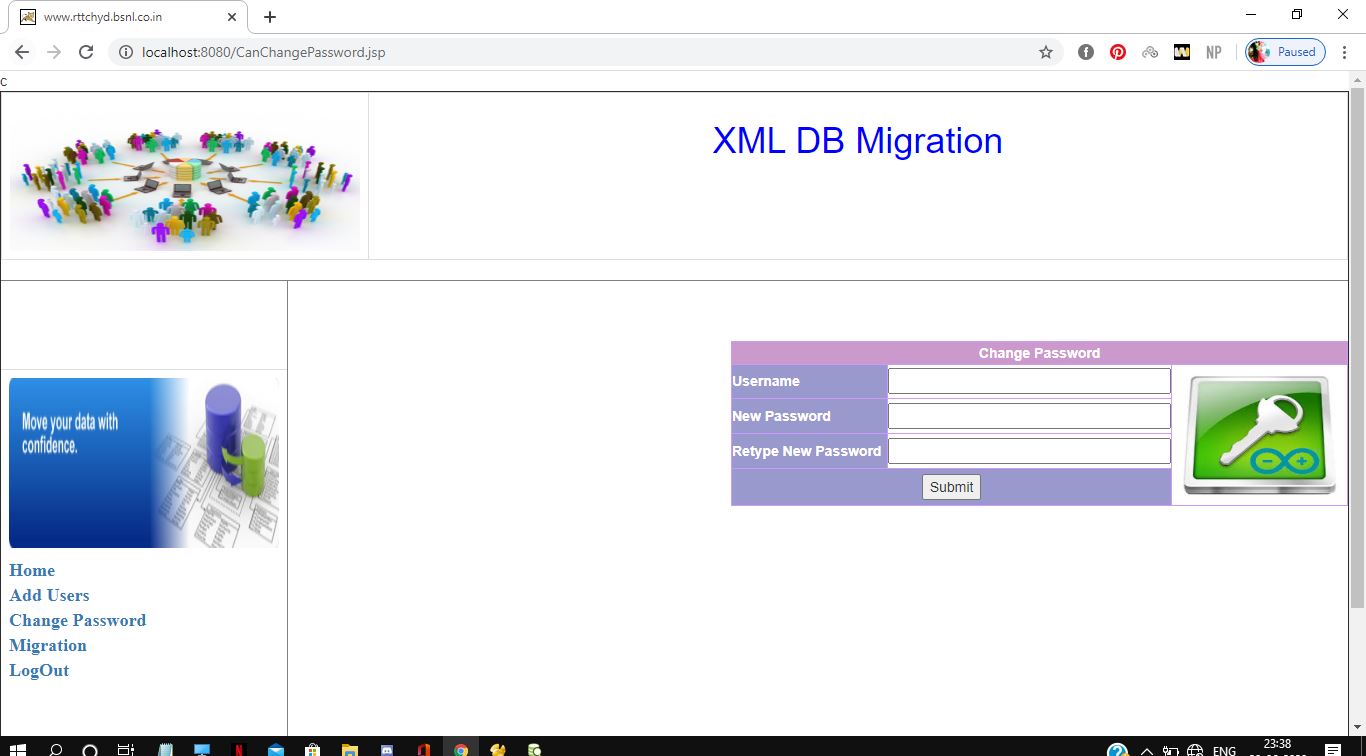
**SCREEN SHOTS**

****

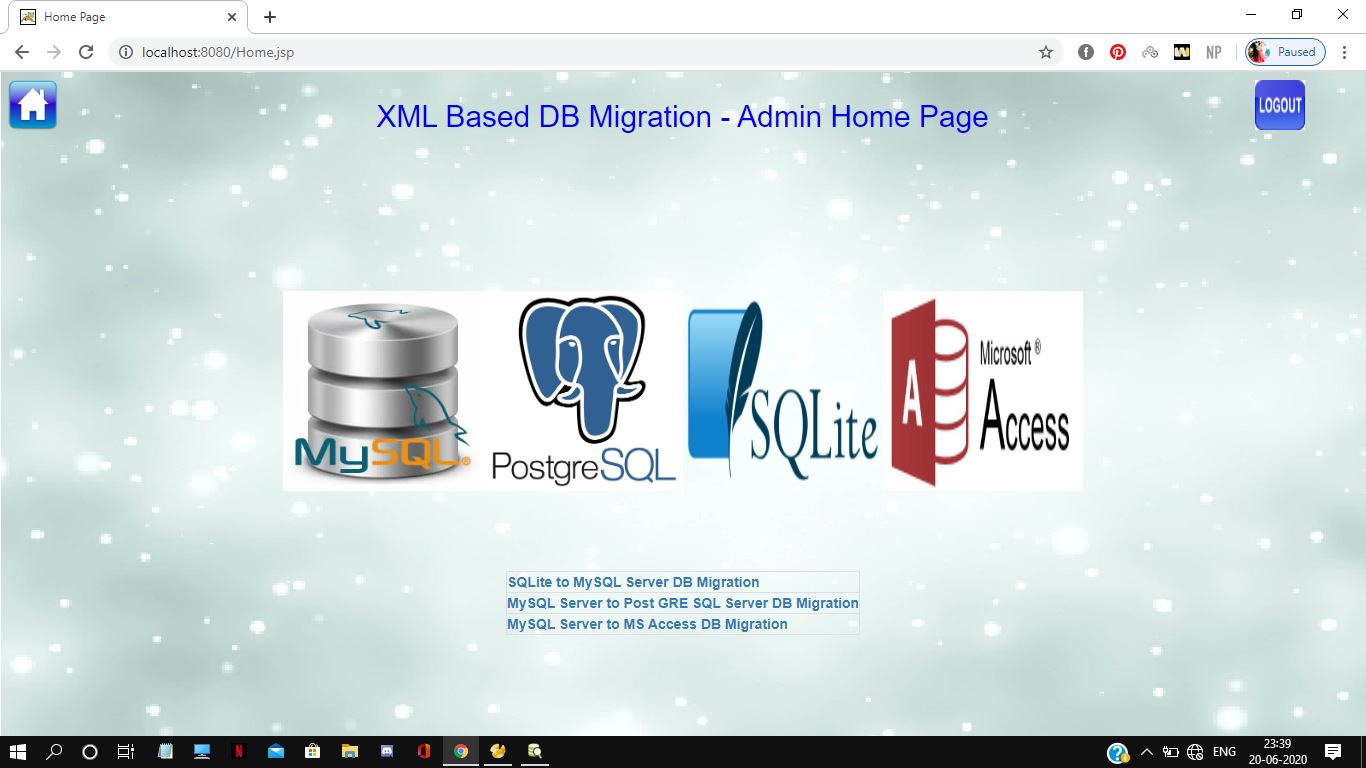
**FIG B1. ADMIN HOME PAGE**

****

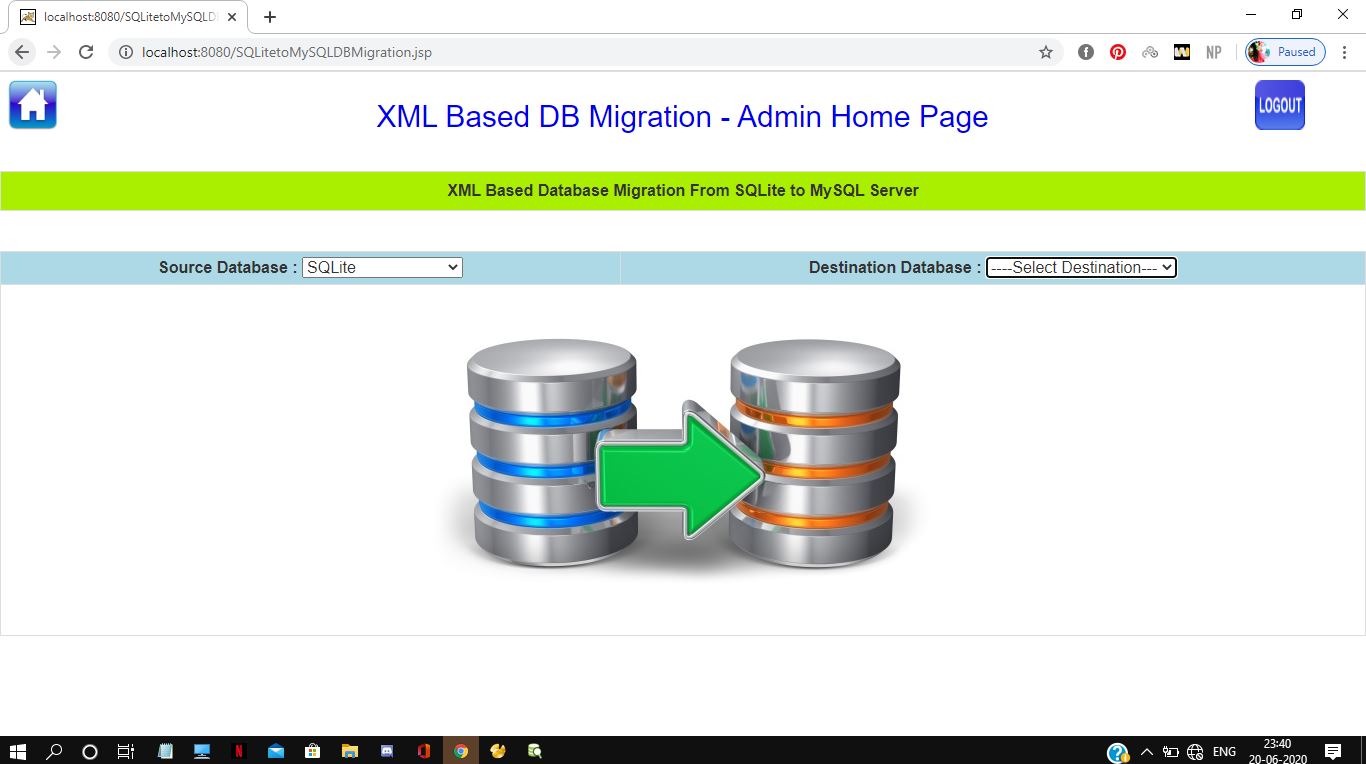
**FIG B2. REGISTRATION PAGE**

****

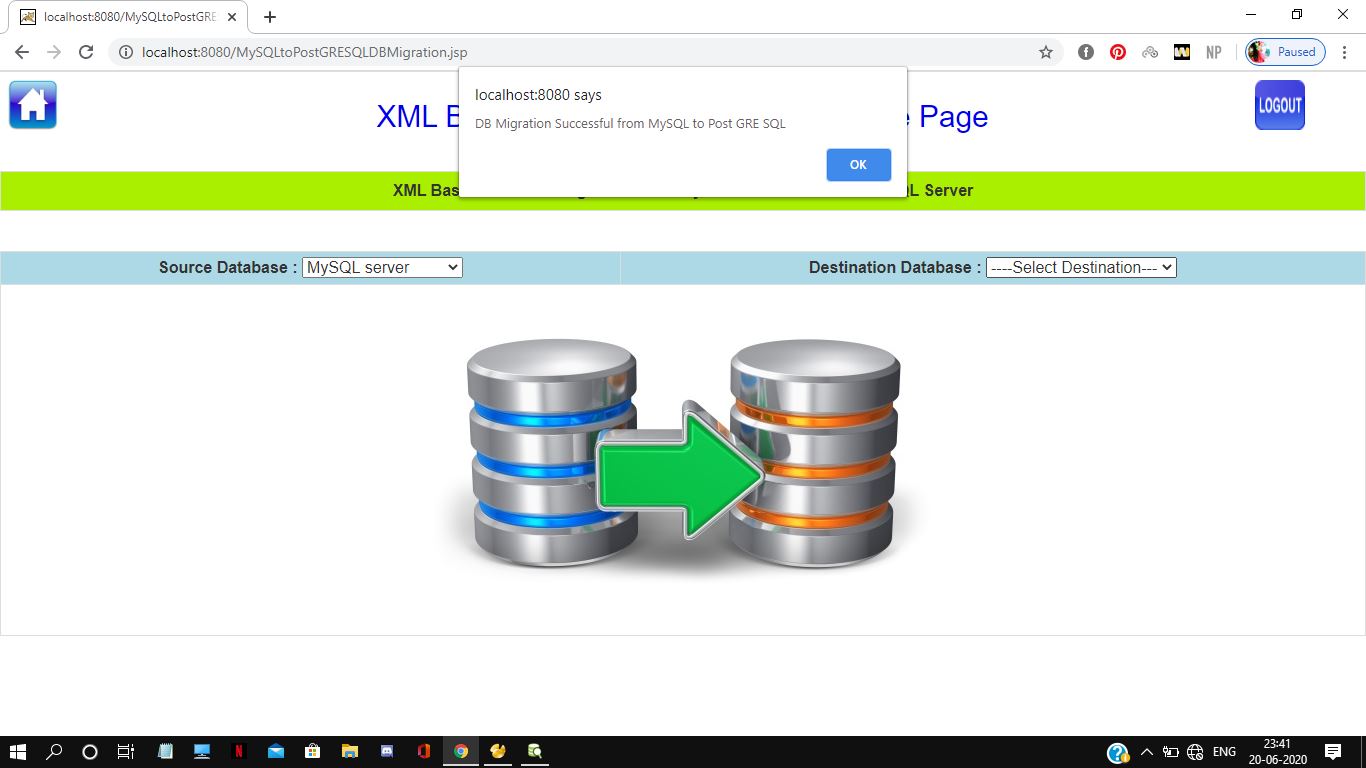
**FIG B3. CHANGE PASSWORD PAGE**

****

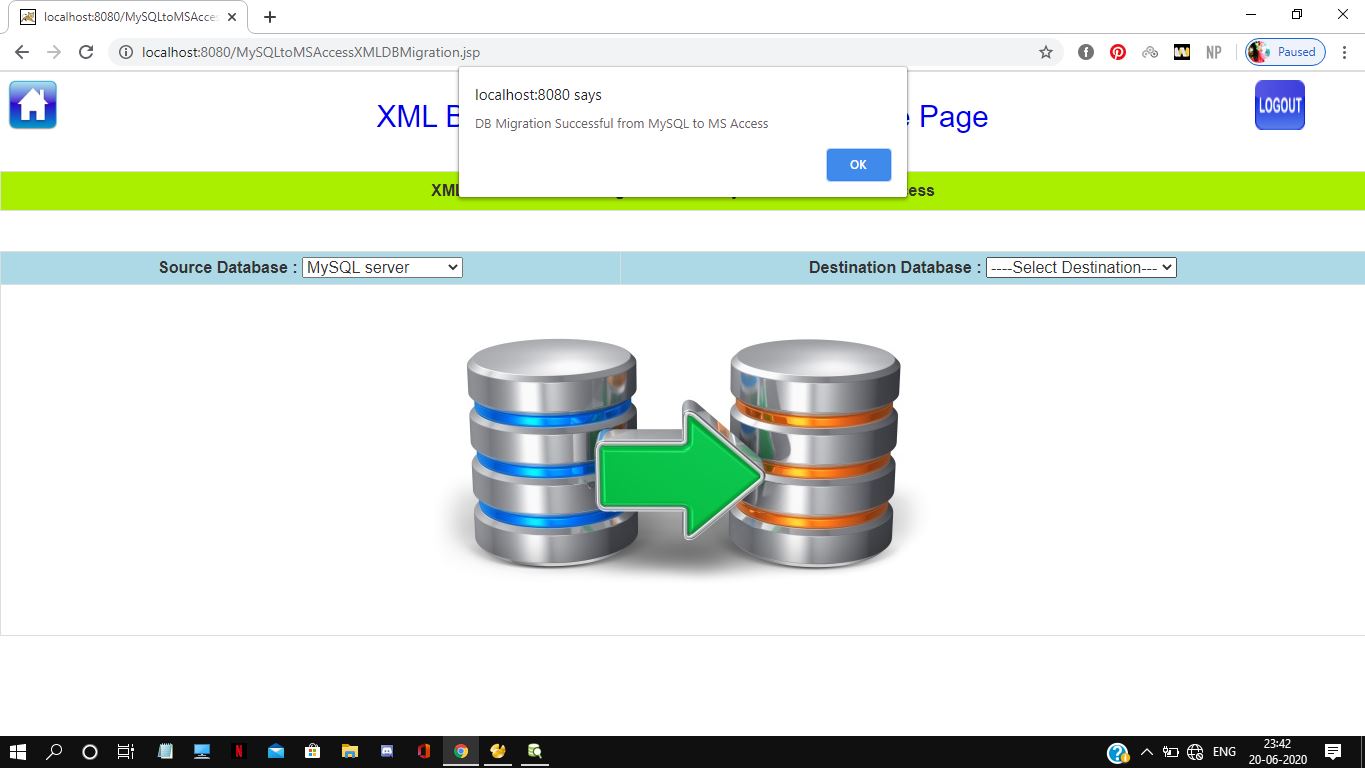
**FIG B4. MIGRATION HOME PAGE**

****

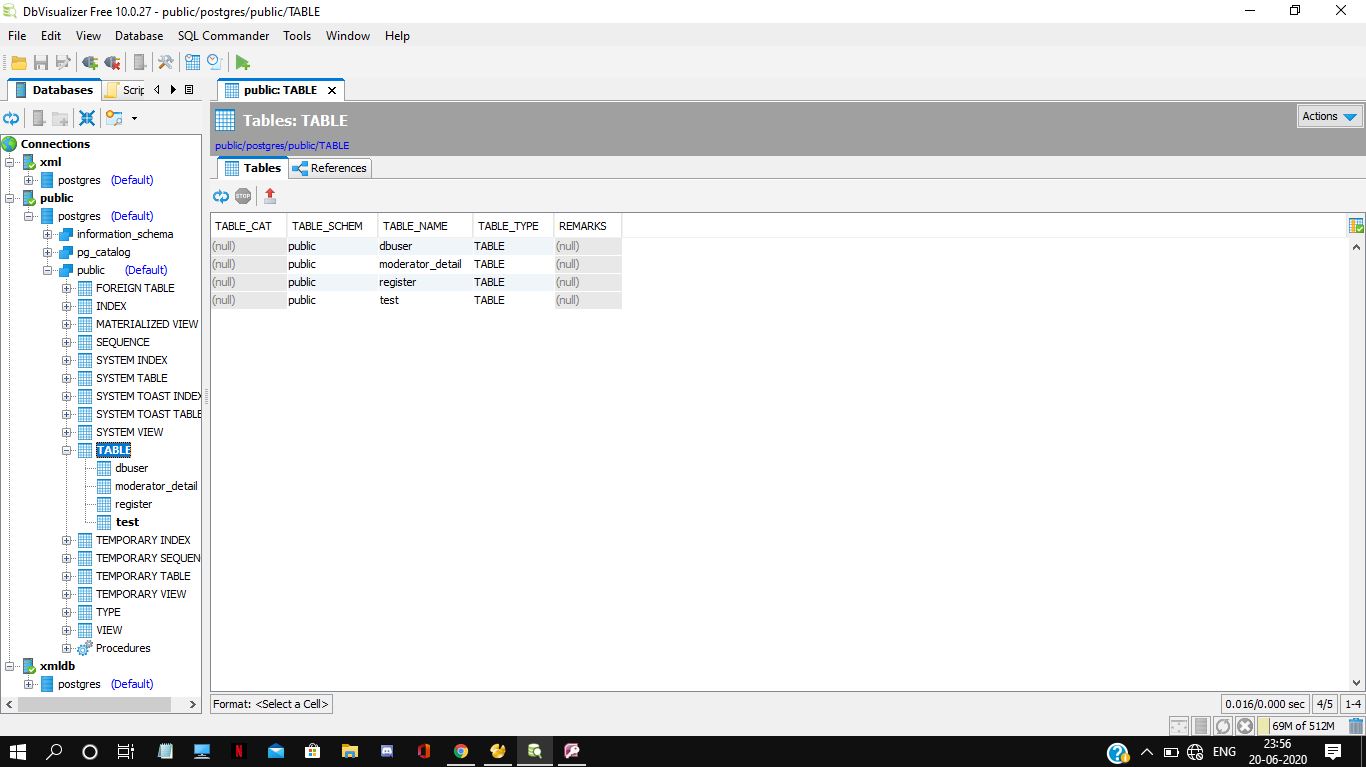
**FIG B5. DATABASE SELECTION BEFORE MIGRATION**

****

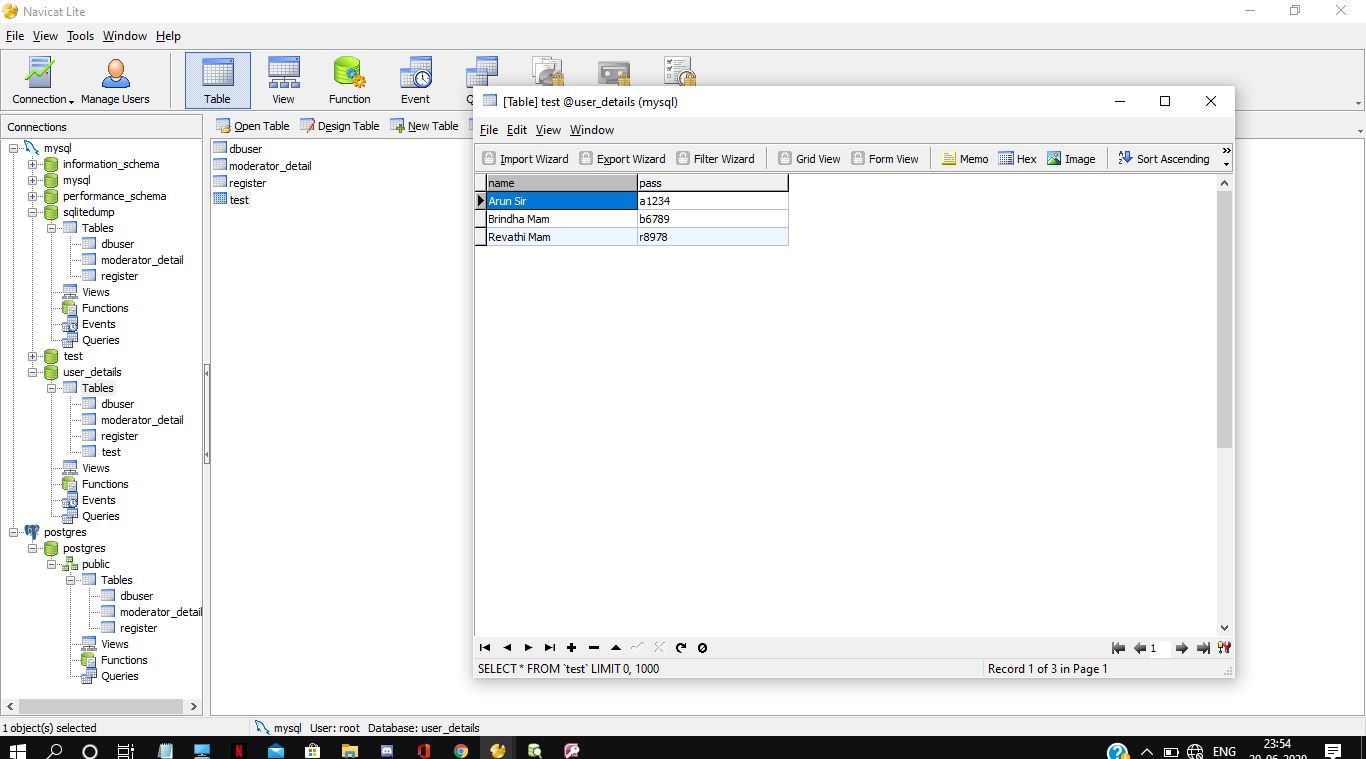
**FIG B6. MIGRATION SUCCESSFUL FROM MYSQL TO POSTGRE SQL**

****

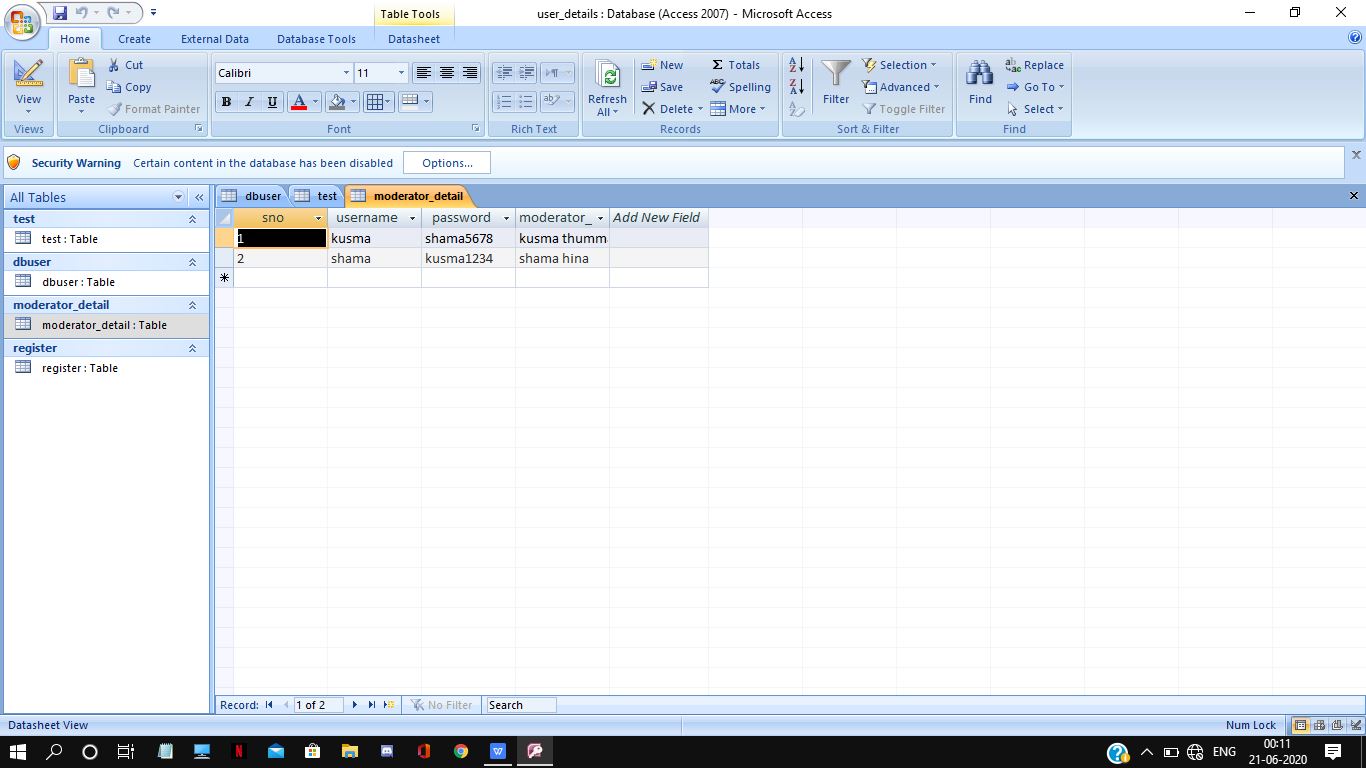
**FIG B7. MIGRATION SUCCESSFUL FROM MYSQL TO MS ACCESS**

****

**FIG B8. POSTGRE SQL PAGE AFTER MIGRATION**

****

**FIG B9. MY SQL PAGE AFTER MIGRATION**

****

**FIG B10. MS ACCESS PAGE AFTER MIGRATION**

**ABOUT THE AUTHORS**

|  |  |
| --- | --- |
| kusma | **KUSMA THUMMAGUNTA** is an Undergraduate student at B.S.Abdur Rahman Crescent Institute of Science and Technology, Chennai, India. She will be completing his Bachelor of Technology (B.Tech) in Computer Science and Engineer in September 2020. She aspires to pursue a career in the domain of Computer Networks, Software Engineering and Web applications. |
| IMG_20191207_233659_024 | **SHAMA HINA.D** is an Undergraduate student at B.S.Abdur Rahman Crescent Institute of Science and Technology, Chennai, India. She will be completing his Bachelor of Technology (B.Tech) in Computer Science and Engineer in September 2020. She aspires to pursue a career in the domain of Software Testing and development , Xml and web Services. |