**Project Portfolio Summary**

*Aakash Chaurasia*

*Computer Science (MCS)*

[*achauras@asu.edu*](mailto:achauras@asu.edu)

All three projects belong to the subject in which I secured at least B grade. All the projects cleared minimum specification of 30% course weightage. Below are the three subject from which I have selected my portfolio project.

* CSE 510: Database Management System and Implementation
* CSE 512: Distributed Database
* CSE 545: Software Security.

**Java Mini Database (CSE 510)**

Database has become an integral part of all the application. It is hard to find any application existing which doesn’t have database. In this coursework we have studied and analyzed the java mini database. This database is implemented in java. It gives practical implementation of database. I studied different concepts throughout the project and studied each and every component in detail. Started studying java mini database concepts. This concept include Page Management Buffer Management, File Management, Disk Management, Indexing, etc. All of this functionalities were implemented using basic java concepts like iterators and interface. Primary objective of this project is to study and extend the java mini database according to our requirement.

In Phase 1 of the project we had to install the database in the linux system. Once we have installed the database, we had to test all the test cases with different scenario. Phase 1 also contained negative test cases which were expected to fail. It also helped us to understand the B+ tree concept and how does it behave with the addition of the nodes.

In Phase 2, we had to implement Page replacement policy. Most Recently Used MRU page replacement algorithm was already implemented. We implemented FIFO, LIFO and LRU-K algorithm for java min db. We also tested this algorithm with different test cases.

In phase 3 we had to experiment and implement the join algorithm. First task of algorithm was to understand relation operators and how does it work, it also demonstrated how nested loop join performed for joining two tables in second half we implemented IEJoin and IESelfJoin algorithm from the given paper. In this phase we created a query plan from the given query and dataset available. Once query plan is finalized using the bushy tree algorithm. Query evaluation is done and result is displayed.

**Geospatial Analysis using Geospark (CSE 512)**

With the increasing amount of data, management of data has become an important part. To crunch the huge number and derive results from them is called big data operations. For analyzing such a huge data, we need to increase the power and capacity of servers, we need to divide the task and handle it separately. In this coursework we have used Apache spark and Hadoop. Hadoop is based on file management it works as a mapper and reducer. In this project we have used geospark, which is specifically designed for performing geospatial queries.

In phase 0 of project, we just setup the Hadoop cluster. For communicating with the different machine we have used ssh less login, which is used to exchange records.

In phase 1 of this project we required to understand the functionality of geospark in details. In this we loaded the csv data in Hadoop file structure. Then we used different types of data structure of geospark. PointRDD is customized JavaRDD for arraylist of two points. Rectangle RDD is JavaRDD for storing arraylist of four points. We have also executed the RangeQuery and Spatial Join Query.

In the phase 2 of this project we have to implement the nested loop join query using the Java RDD and geospark data structure. Using algorithm we have implemented the join query. Then we evaluated this join query with other available functions which are available. For comparison of performance we have used ganglia which gives graphical representation for comparing performance.

In the final phase we have to identify the 50 hotspots in New York from data extracted new York yellow cab data. We have used different data structure of geospark and came up with the algorithm to find out hotspots. This hotspots were identified on the basis of Gi\* which is calculated on the basis of neighbors.

**Prevention of SQL Injection (CSE 545)**

This coursework helped me to understand different types of attacks which are possible. I learned different types of vulnerabilities existing in system, network level, physical level, application level, http level IP level, MAC level etc. In this coursework I worked on project call Prevention of SQL Injection. Nowadays information is wealth, if information is leaked it seems bad faith for company. SQL injection is one of the prime attack possible on database. So in order to detect and avoid SQL injection we have created a PHP library which helps in this. Usually sql injection can be exploited from the simple bug in the code. This bug can be exploited to steal information. To avoid such kind of mistakes coder has to use our PHP library. This library eliminates SQL injection attacks which are available.

First Order SQL is attack in which input is given in such a way that sensitive data is displayed for which attacker is not authorized. Second order attack deals with modification of database records in order to exploit it for the future attack. In this user modify the database legit way, then exploit that to extract information based on that.

We had identified all different kinds of patterns available for sql injection and handle it using the PHP code, hence sanitizing the input data from exploiting vulnerabilities and retrieving results.