**AUTOMATION OF ENTERPRISE BATCH JOB EXECUTION MONITORING AND REPORT THROUGH CUSTOMIZABLE WEB APPLICATION**

*A Project*

*submitted in partial fulfillment of the requirements for*

*the award of the Degree of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

By

**OSHIN BANERJEE**

**(BE/3019/2015)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**BIRLA INSTITUTE OF TECHNOLOGY**

**MESRA-835215, RANCHI**

**ALLAHABAD CAMPUS**

**2019**

**ACKNOWLEDGEMENT**

We owe our deepest gratitude to our guide **Projjwalanka Pramanick,** who helped us throughout the project. He made sure that we learnt by practice, always motivating us to think a step further, work a little harder. Our interactions with him always resulted in newer ideas and proved beneficial towards our work. Without his constant presence and supervision our work would not have been successful. We especially acknowledge the many useful discussions we had among ourselves that helped us understand some of the subtle technical problems in a better way.

We would also like to thank **Krishnendu Ganguly** (Manager of Cognizant Academy) for guiding us throughout the entire period, helping us and encouraging us to give our best.

Thank you one and all.

**OSHIN BANERJEE**

**(BE/3019/2015)**

**CONTENTS**

|  |  |
| --- | --- |
| **S.No.** | **TOPICS** |
| 1. | Introduction |
| 2. | Iterative Waterfall Model |
| 3. | System Architecture |
| 4. | Technology Stack |
| 5. | Hardware requirements |
| 6. | Project Description |
| 7. | Application Features |
| 8. | Database Design |
| 9. | Screenshots |

INTRODUCTION

A batch job is a computer program or set of programs processed in batch mode. This means that a sequence of commands to be executed by the operating system is listed in a file (often called a batch file, command file, or [shell](https://kb.iu.edu/d/agvf) script) and submitted for execution as a single unit.

In many cases, batch jobs accumulate during working hours, and are then executed during the evening or another time when the computer is idle. This is often the best way to run programs that place heavy demands on the computer.

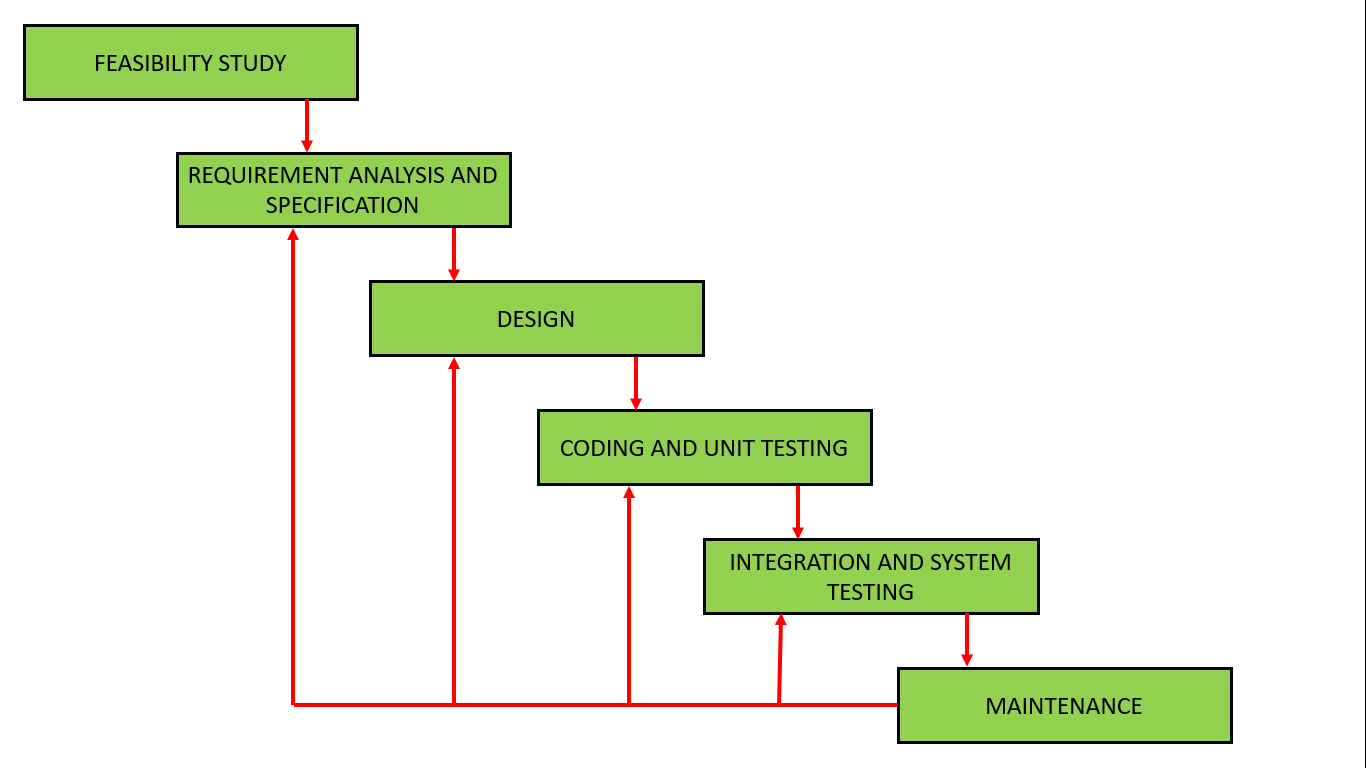
Batch jobs are backbone of any enterprise’s IT eco system. A large scale enterprise can consist of more than 100 applications and must be dealing with huge volume of transactions every day. The role of batch jobs (Unix, ETL, Windows batch etc.) are essentials specially for inter application data migration/ transformation and for summarization and reporting. There may be more than several thousand batch jobs running in several servers round the clock. Monitoring and troubleshooting batch job execution is a huge task which requires significant amount of manual and tedious effort. There are direct business implications if critical batch jobs fail and side effects are not mitigated on time.

The proposed web application aims to reduce manual effort to monitor batch jobs and generates alerts when required. The application shall be able to eliminate human intervention for very common types monitoring tasks such as timely start of batch job, failure, delay and potential error during execution of batch jobs. This application can be configured to monitor batch jobs running in different serves across platforms. Error detection technology should be primarily based on static log analysis. The application shall have reporting and summarization features as per user’s requirement.

**ITERATIVE WATERFALL MODEL**

**The iterative waterfall model provides feedback paths from every phase to its preceding phases, which is the main difference from the classical waterfall model.**

Feedback paths introduced by the iterative waterfall model are shown in the figure below.



When errors are detected at some later phase, these feedback paths allow correcting errors committed by programmers during some phase. The feedback paths allow the phase to be reworked in which errors are committed and these changes are reflected in the later phases. But, there is no feedback path to the stage – feasibility study, because once a project has been taken, does not give up the project easily.  
It is good to detect errors in the same phase in which they are committed. It reduces the effort and time required to correct the errors.

**Phase Containment of Errors:** The principle of detecting errors as close to their points of commitment as possible is known as Phase containment of errors.

**Advantages of Iterative Waterfall Model**

* **Feedback Path:** In the classical waterfall model, there are no feedback paths, so there is no mechanism for error correction. But in iterative waterfall model feedback path from one phase to its preceding phase allows correcting the errors that are committed and these changes are reflected in the later phases.
* **Simple:** Iterative waterfall model is very simple to understand and use. That’s why it is one of the most widely used software development models.

**Drawbacks of Iterative Waterfall Model**

* **Difficult to incorporate change requests:** The major drawback of the iterative waterfall model is that all the requirements must be clearly stated before starting of the development phase. Customer may change requirements after some time but the iterative waterfall model does not leave any scope to incorporate change requests that are made after development phase starts.
* **Incremental delivery not supported:** In the iterative waterfall model, the full software is completely developed and tested before delivery to the customer. There is no scope for any intermediate delivery. So, customers have to wait long for getting the software.
* **Overlapping of phases not supported:** Iterative waterfall model assumes that one phase can start after completion of the previous phase, but in real projects, phases may overlap to reduce the effort and time needed to complete the project.
* **Risk handling not supported:** Projects may suffer from various types of risks. But, Iterative waterfall model has no mechanism for risk handling.
* **Limited customer interactions:** Customer interaction occurs at the start of the project at the time of requirement gathering and at project completion at the time of software delivery. These fewer interactions with the customers may lead to many problems as the finally developed software may differ from the customers’ actual requirements.

**SYSTEM ARCHITECTURE**

A **system architecture** or **systems architecture** is the conceptual model that defines the structure, behavior, and more views of a **system**. An **architecture** description is a formal description and representation of a **system**, organized in a way that supports reasoning about the structures and behaviors of the **system**.

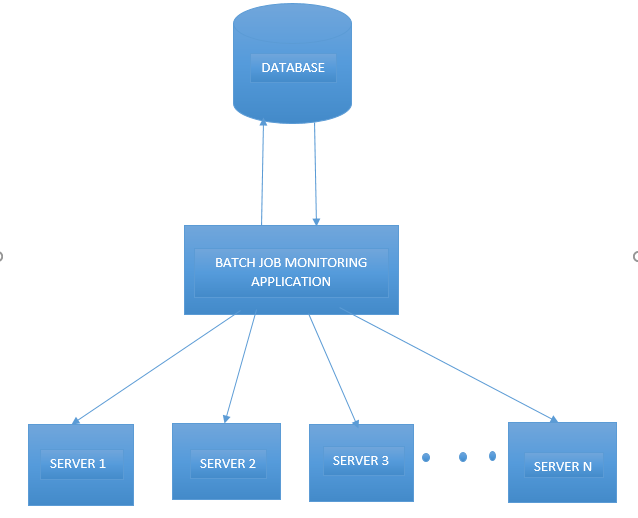
* SOFTWARE ARCHITECTURE

One of the types of system architecture is Software Architecture. **Software architecture** refers to the high level structures of a [software system](https://en.wikipedia.org/wiki/Software_system) and the discipline of creating such structures and systems. Each structure comprises software elements, relations among them, and properties of both elements and relations. The *architecture* of a software system is a metaphor, analogous to the [architecture](https://en.wikipedia.org/wiki/Architecture) of a building. It functions as a blueprint for the system and the developing project, laying out the tasks necessary to be executed by the design teams. One of the design styles of software architecture is **client-server architecture.**

* CLIENT SERVER ARCHITECTURE

The client-server model of computing is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters called clients. Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server machine is a host that is running one or more server programs which share their resources with clients.

Client-server software architecture is versatile and flexible in today’s fast-changing IT landscape. It is modular in structure and relies on messaging services for communication between components. They were designed to improve flexibility, usability, scalability, and interoperability. Software flexibility implies the ability for a program to change easily according to different users and different system requirements.



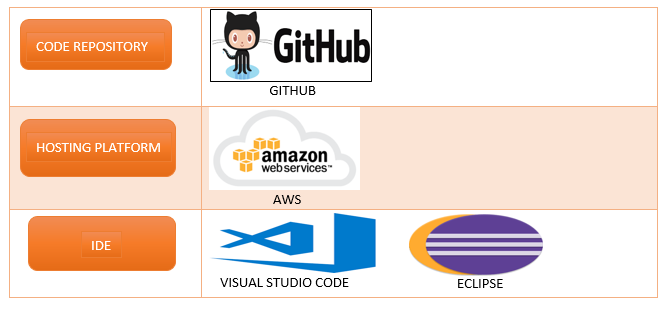
The batch job monitoring application resides in cloud which acts as the master server which allows the client servers **server1,server2,…,server n** to share the application, without any affect on the performance.

As we can see, the batch job monitoring application interacts with the database in order to exchange data to and fro as and when required. The user creates and enters the batch and schedule details into the application. The application stores the data into the database.

The application retrieves the server name from the database in order to monitor the mentioned batch job. It continues to alert the user regarding the current status of the batch job while also updating its database.

The application also facilitates generating summarization and report as per the user’s requirements.





**HARDWARE REQUIREMENTS:**

* RAM: 4GB and above
* Processor: Intel Core i3 and above
* Internet Connection
* Operating System: Windows 7 and above, Unix, Mac OSX 10.8 and above
* Hard Disk: 20GB and above

**BROWSERS:**

* Chrome\* 36+
* Edge\* 20+
* Mozilla Firefox 31+
* Internet Explorer 11+ (Windows only)
* Safari 6+ (Mac OS only)

**PROJECT DESCRIPTION**

The web application “**Automation of Enterprise Batch Job Execution Monitoring and Report through Customizable Web Application**” allows the user to add batch jobs which they wish to monitor to the database. The user also enters the schedule details of the batch job such as job start date, end date, run days, etc. After the batch job details and its schedule details is updated into the database, the web application keeps a track of primary problems such as whether the batch jobs start on time, its end time exceeding the maximum days, reports if the batch jobs are starting on the mentioned run days, etc.

Apart from the above primary issues, it also keeps a track of the current status of the batch jobs. This task would otherwise had to be carried out by a human, which is time consuming as well as tedious. There might be hundreds of batch jobs which require monitoring. The application designs a dashboard that displays which batch jobs are active, inactive, completed or failed. Although it may seem that we will know that a batch job has failed when an application does not runs properly, many times failure of some critical batch jobs might have direct business implications. Rectifying and getting such batch jobs run on time can save both time and money of an organization. Since the function of the user is only for entering the batch job and schedule details, the chances of error are minimal as well as there is no need for a constant watch on batch job execution as the web application will alert the user whenever a batch job starts, completes or has failed.

**APPLICATION FEATURES:**

* BATCH JOB DETAILS PAGE:

The user needs to enter **job id**, **job name**, **executable name**, **executable path, log file name**, **log file location** and **description**. All the fields are compulsory except description, which is optional. In case any of the compulsory fields is not entered, an error is displayed below the required field. Each of the fields have their own constraints as to minimum and maximum character limit, allowed characters, etc. If any of the constraints is not met, error is displayed for the particular field. Once all constraints have been satisfied, the data from this page is updated to database table **in\_batch\_details**.

* SCHEDULE DETAILS PAGE:

For each batch job entered into table **in\_batch\_details,** a schedule needs to be created. Here, we need to enter the **job id**, **server name**, **executable name**, **maximum duration(in days), dependency, run days(in days), start time** and **end time**. The constraints for each field needs to be satisfied before submission. The data from this page is updated to database table **in\_schedule\_details** where **job id** is the foreign key from **in\_batch\_details** table. The job id is selected from a list of job id’s that are in **in\_batch\_details** table.

**ADVANTAGES:**

* No human intervention required for tracking batch jobs.
* No need to remember when a batch job should start or end, as the web application will alert the user whenever a batch job starts or ends.
* Provides a summarized report on batch job execution according to the user’s requirement.

**DISADVANTAGES:**

* Requires the user to enter the batch and schedule details. May result in unwanted error such as spelling mistakes, etc.

DATABASE DESIGN

**TABLES**:

TABLE: in\_batch\_details

|  |  |  |
| --- | --- | --- |
| S.NO. | COLUMN NAME | VALUE TYPE |
| 1. | Job\_ID | varchar(10) |
| 2. | Job\_Name | varchar(30) |
| 3. | Description | varchar(200) |
| 4. | Executable\_Name | varchar(35) |
| 5. | Executable\_Path | varchar(100) |
| 6. | Log\_File\_Name | varchar(20) |
| 7. | Log\_File\_Location | varchar(100) |
| 8. | Date\_Time\_Created | timestamp |
| 9. | Created\_By | varchar(20) |
| 10. | Date\_Time\_Modified | timestamp |
| 11. | Modified\_By | varchar(20) |

Primary Key: Job\_ID

TABLE: in\_schedule\_details

|  |  |  |
| --- | --- | --- |
| S.NO. | COLUMN NAME | VALUE TYPE |
| 1. | Server\_Name | varchar(20) |
| 2. | Batch\_ID | varchar(10) |
| 3. | Run\_Days | varchar(10) |
| 4. | Start\_Time | varchar(10) |
| 5. | End\_Time | varchar(10) |
| 6. | Max\_Duration | int(11) |
| 7. | Active\_Indicator | varchar(10) |
| 8. | Date\_Time\_Created | timestamp |
| 9. | Created\_By | varchar(20) |
| 10. | Date\_Time\_Modified | timestamp |
| 11. | Modified\_By | varchar(20) |
| 12. | Dependency | varchar(255) |

Primary Key: Server Name

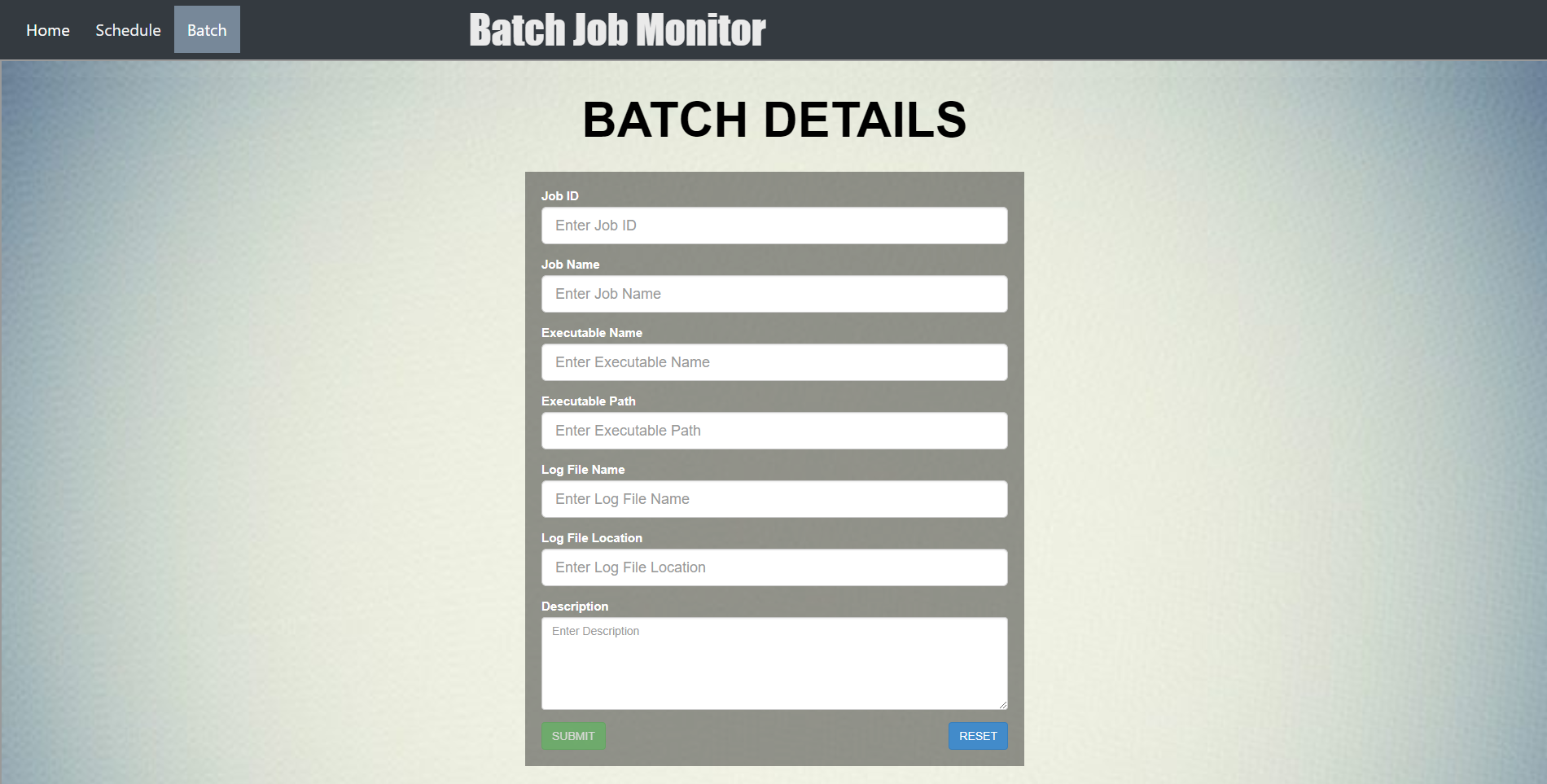
Foreign Key: Batch\_ID references in\_batch\_details(Job\_ID)

TABLE: in\_execution\_summary

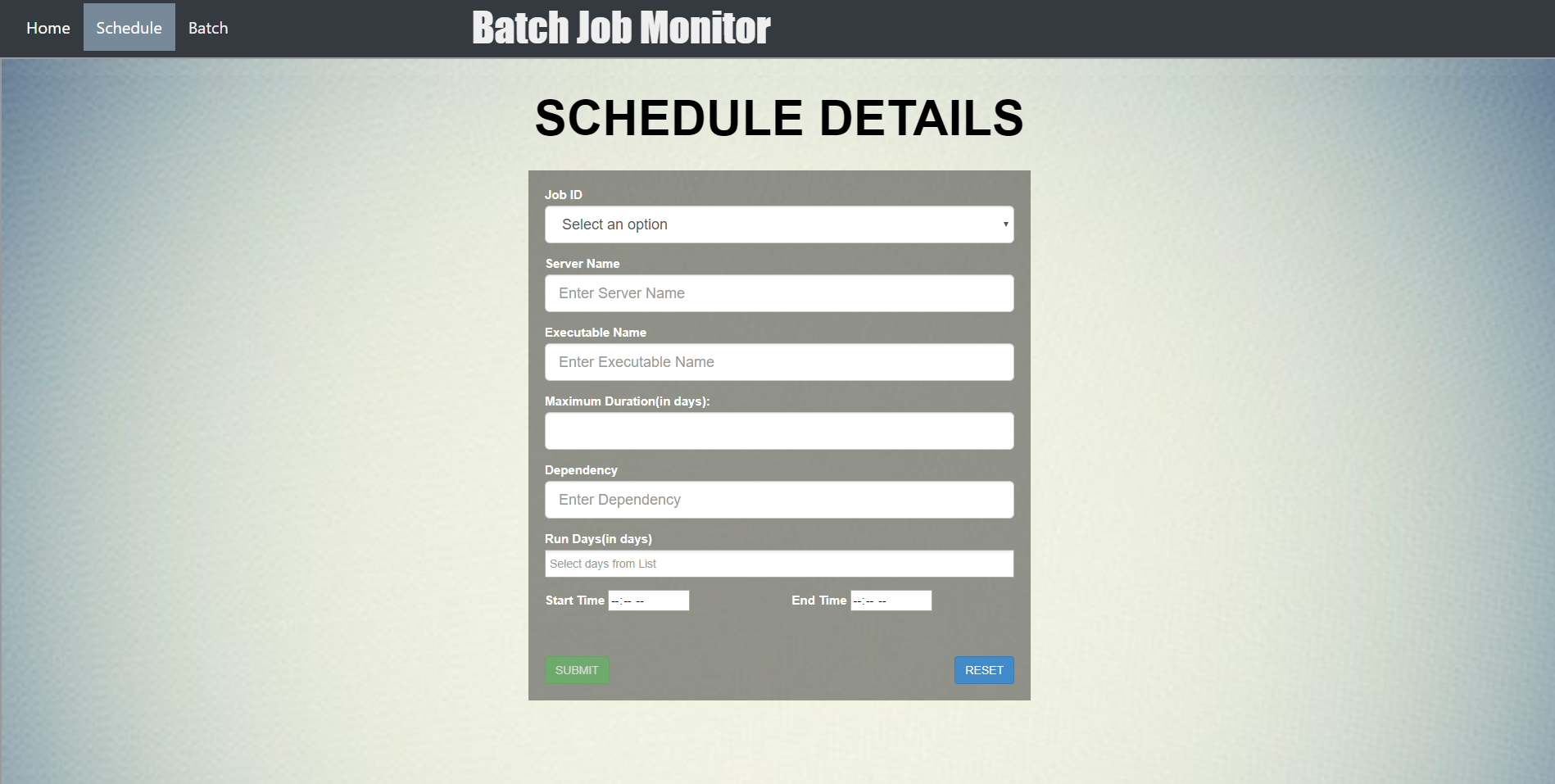
|  |  |  |
| --- | --- | --- |
| S.NO. | COLUMN NAME | VALUE TYPE |
| 1. | Server\_Name | varchar(20) |
| 2. | Batch\_ID | varchar(10) |
| 3. | Schedule\_Start | varchar(10) |
| 4. | Actual\_Start | varchar(10) |
| 5. | Schedule\_End | varchar(10) |
| 6. | Actual\_End | varchar(10) |
| 7. | Execution\_Status | varchar(10) |
| 8. | Date\_Time\_Created | timestamp |
| 9. | Date\_Time\_Modified | timestamp |

**SCREENSHOTS**

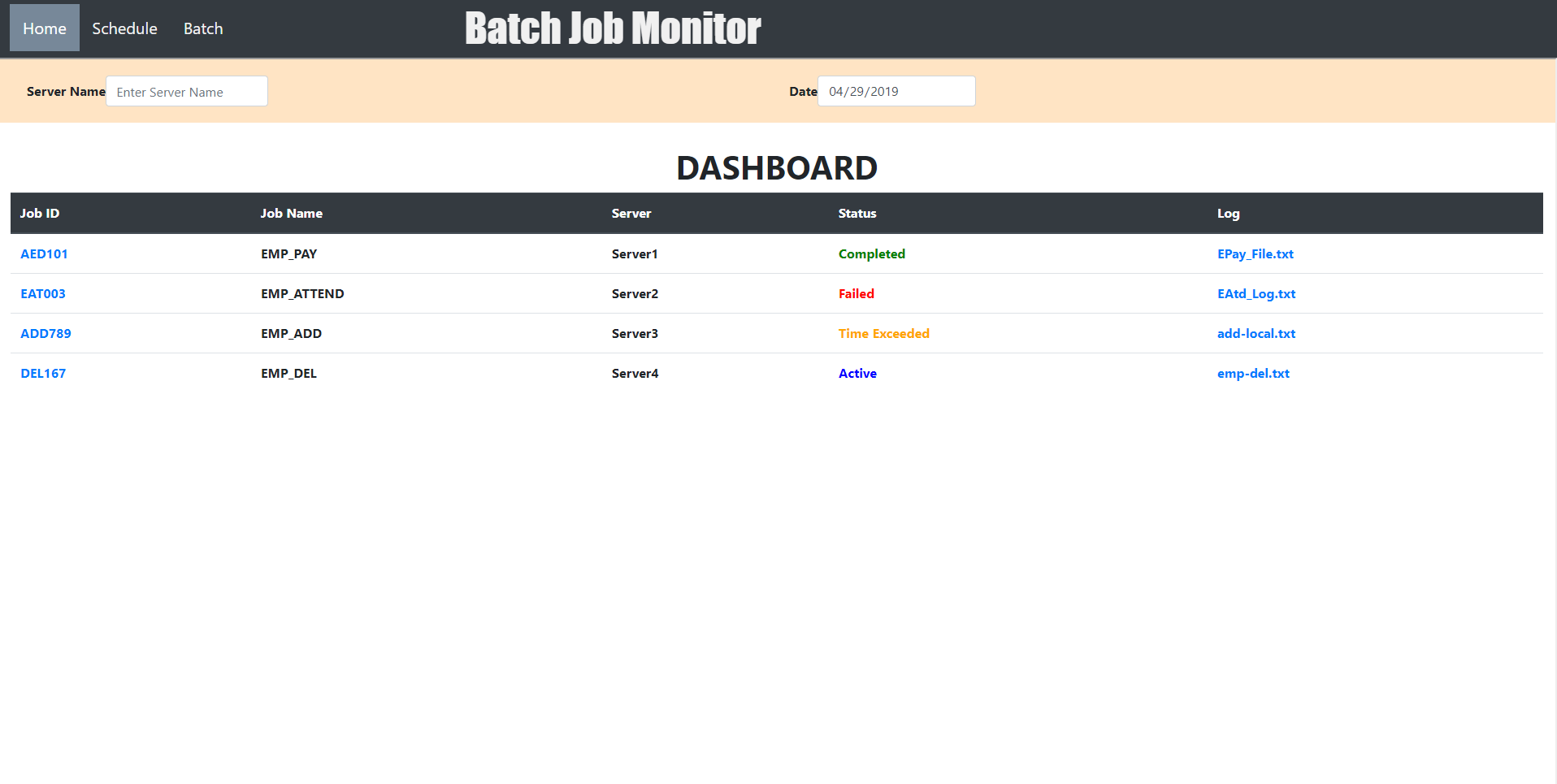
**ADD BATCH PAGE**

****

**ADD SCHEDULE PAGE**



**HOME PAGE**



FUTURE SCOPE

* Advanced module to take corrective action in case of batch failure:

Corrective action is an aspect of quality management that aims to rectify a task, process, product, or even a person’s behavior when any of these factors produce errors or have deviated from an intended plan. Corrective actions can be thought of as improvements to an organization to eliminate undesirable effects.

In our application, advanced module can be created which can handle errors and restart faulty batch jobs on their own. This would significantly reduce human effort as well as the time wasted waiting for rectification of errors. This will benefit the organization in many ways, as the reduction in response time to errors for critical jobs will save resources which can be utilized for other tasks.

* More advanced alert system to reduce turn around time: Turnaround time is the total time taken between the submission of a program/process/thread/task for execution and the return of the complete output to the customer/user.[[3]](https://en.wikipedia.org/wiki/Turnaround_time#cite_note-3) It may vary for various programming languages depending on the developer of the software or the program. Turnaround time may simply deal with the total time it takes for a program to provide the required output to the user after the program is started.

An organization may follow many types of topologies: star,bus, mesh, etc. Whichever topology is followed, it may never be guaranteed that the estimated turn around time will be same as actual turn around time. The cause for such delay may be numerous, yet when certain critical tasks fail to respond on time, it may result in huge lose for the organization, even if it is for a picosecond. As mentioned earlier, our application alerts the user about the status of the batch jobs executing. Since, our application resides on the server, its turn around time may also lag. Adopting advanced alert systems such as mails, sms on phone, etc, might help in reducing such significant lags in turnaround time.

* AI based predictive maintenance module to predict failure cases and take preventive action:  Preventive actions are implemented prior to the occurrence of a non-conformance or deviance. Preventive action programs are proactive tools tied to quality system elements that involve monitoring and assessing system and process effectiveness. Being proactive means staying on top of process inputs, outputs, and other data that signal[trends](http://www.sciqual.com.au/sites/default/files/corrective_vs.pdf) and abnormalities and acting on them before they become issues.

With the help of AI based predictive maintenance module, we can predict the possible cases under which our batch jobs may fail. Predicting such errors in advance and taking preventive action can significantly reduce the time and effort in running and rectifying a faulty batch job.

**CONCLUSION**

Maintaining and keeping an eye on BATCH jobs can be a daunting task. Even if the faulty batch job is found, locating its log file is another enormous task. Considering that an enterprise may contain more than 1000 batch jobs, it is highly possible that atleast 10 jobs might be faulty. This may cause abrupt ending of applications or may give anomalous results. If such cases happen frequently, this might result in huge loss on the part of the organization.

Our project aims at reducing such cases to the very least. It keeps a track of the current status of the batch jobs, alerting the user about the same. This reduces the effort on the part of the user, to fetch faulty jobs and the occurrence of errors. The application also maintains a record of the log file generated, for each batch job which makes it easier on the part of the user to view the log details as and when required. Schedules containing server name, start date, end date, etc. are also maintained by the application which frees the user from the task of remembering the schedule details of the batch job.