

# Automotive Multifaceted Result Analyzer

Mr. Rupesh Navnath Vittekar, Prof. Priya M. Shelke

*Department of Information Technology,  
Vishwakarma Institute of Information Technology,  
Savitribai Phule Pune University, India.*

*Email: [rupesh.vittekar2010@gmail.com](mailto:rupesh.vittekar2010@gmail.com), [priya.shelke@viit.ac.in](mailto:priya.shelke@viit.ac.in)*

**Abstract**—In Engineering College each department needs to perform result analysis. In this process most important part is to convert the result from PDF to EXCEL format. Which is done manually by the faculty and it takes more time to generate the reports. Our proposed work will be used to show the future vision clearly, which specifies that where the result is lagging and where is the scope for improvement. This report will be very easy to understand and it will clearly state the progress of each department. Disadvantages of manual process are that, it requires more time to develop and it includes manual effort which leads to the complexity. Our Proposed work is efficient to avoid these disadvantages by developing a web system which converts the University result in the EXCEL format. In this web application we reduce the complexity of the manual process. Main advantage of our proposed work is to reduce the time and manual work. Our proposed application will generate several reports like a grade wise report, final report, subject report in excel format.

## I. INTRODUCTION

Nowadays in Engineering College each department needs to perform Result analysis for analysing student performance. The analysis is done manually by the faculty and it takes more time to generate the report. To ease this lengthy work we are developing the system that will help the faculty in result analysis. This system will scan the documents for the required patterns and save the result in desired format. This will also ease the process of report generation. Different types of report formats are also available in the system which will help the staff in individual evaluation of result.

This software system will be a Result Analysis System for a college faculty who analyses the University result and generates analysis report. This system will be designed to maximize the faculty's productivity by providing tools to assist in automating the result analysis and report generating process. This system will meet the Institute's needs and it is easy to understand and use. Morespecially, this system is designed to allow the faculty to manage and analyse the University result of the students and generate report which can be stored in the Institute database. The software will facilitate to store the University result of each semester in the Institute database which can be used in future. Also it helps to generate the grade wise reports, subject wise reports and overall final reports.

### 1. Need :

Aim of this work is to develop a highly useable Web based report generation (Excel Format) system for University Result of Engineering student for different branches. It is the requirement of Every Engineering College to analyse College performance or specifically department performance. Result

analysis is most important task and this system will be used to analyse that result and generate the some specific required reports which is used in the Evaluation (performance). The idea about this project is in progress.

### 2. Basic Concepts :

This project is expected to bring in functionalities that would automate the processes in the Result Analysis.

### 3. Application:

System will enable faculty to generate the reports analysis (Excel and Graphical Format). System will automate all the processes in the Result Analysis Filed.

## II. PROBLEM DEFINITION

The objective of Automotive Multifaceted Result Analyzer System is to provide Reports (Tabular and Graphical Format) against University Result. In my research, the Result is in the pdf format and it changes when the University pattern changes so every time result pdf is different than older pdf. So that reading the result which is in pdf format is difficult task.

Our research work divided into 4 steps they are as follows;

1. The system is making use of Regular Expression to recognize the result pattern.
2. Read that pattern to fetch the data from the pdf format.
3. Save the data in the Database.
4. Generate the Automated Reports.

These four steps are use to overcome the disadvantages of previous system. This research is mainly focused on analysing the student result and their Performance by using automated system. This helps to show the result in Standard format (Excel format).

Our system is generating following report in excel format;

1. Grade wise report.

2. Subject wise report.
3. Final Report.
4. Top 5 rankers report.

### III. LITERATURE REVIEW

Regular expressions are extremely useful technique for working with text. Whether you need to validate user input, search for patterns within strings, or reformat text in powerful ways and Support for regular expressions in the Microsoft®.NET Framework is first-class, and even just within Microsoft® ASP.NET there are controls that rely on the language of regular expressions [1].

To extract data from PDF file we require regular Expression and *.regex* class is use for evaluation. The quantifier's terminology provides a simple way to specify within a pattern that how many times a particular character or set of characters is allowed to repeat itself. Quantifiers always refer to the pattern immediately proceeding (to the left of) the quantifier, which is normally a single character unless parentheses are used to create a pattern group [1].

To constructs within regular expressions that have special meaning are referred to as metacharacters such as the \*,?, +, and { } characters. Several other characters have special meaning within the language of regular expressions. These include the following: \$ ^ . [ ( ) ] and \ [1].

Character classes are a mini-language within regular expressions defined by the enclosing hard braces [ ]. The simplest character class is simply a list of characters within these braces, such as [aeiou]. When used in an expression, any one of these characters can be used at this position in the pattern [1].

Patteren	Input (Matches)
^b[aeiou]t\$	Bat, bet, bit, bot, but
(abc){2,3}	abcabc, abcabcabc

ASP.NET provides a suite of validation controls, which make validating inputs on web forms extremely easy compared to the same task using legacy (or *classic* if you prefer) ASP. One of the Regular Expression Validator allows you to validate inputs by providing a regular expression which must match the input [1].

The *regex* class represents the .NET Framework's regular expression engine. It can be used to quickly parse large amounts of text to find specific character patterns; to extract, edit, replace, or delete text substrings; and to add the extracted strings to a collection and to generate a report. A regular expression is used commonly for validating, matching, and replacing. In many cases, these can be accomplished using static methods of the *regex* class, without any need to instantiate the *regex* class itself. To perform validation, all you must do is create or find the right expression and apply it to your input string using the *IsMatch()* method of the *regex* class [2].

To view reports, analyzed results are deployed to a report server or it exists on the local file system. We can use the WebForms ReportViewer control to render them in a Web application [3].

Microsoft Visual Studio 2008 provides two ReportViewer controls for integrating report viewing functionality into your applications. There is a version for Windows Forms-based applications and one for Web Forms applications. Each control provides similar functionality but each is designed to target their individual environments. Both controls can process reports that have been deployed to a report server (remote processing mode) or have been copied to a computer where Microsoft SQL Server Reporting Services has not been installed (local processing mode) [3].

### IV. TECHNOLOGY USED

#### REGULAR EXPRESSION:

A regular expression is a pattern that could be matched against an input text. The .Net framework provides a regular expression engine that allows such matching. A pattern consists of one or more character literals, operators, or constructs. Support for regular expressions in the Microsoft.NET Framework is first-class, and even just within Microsoft ASP.NET there are controls that rely on the language of regular expressions. A regular expression is a pattern that could be matched against an input text. The .NET framework provides a regular expression engine that allows such matching. A pattern consists of one or more character literals, operators, or constructs.

Regular Expression required *regex* class in .NET Regular expressions as they exist today were invented in the 1950s. Regular expressions were originally used to describe "regular sets," which were patterns under study by neurophysiologists. Credit for the first regular expressions is given to the mathematician Stephen Kleene. Eventually, Ken Thompson built support for regular expressions into *qed* and *grep*, both very popular text utilities.

#### A. REGEX CLASS:

This class is used to represents an immutable regular expression and for representing a regular expression. It has the following commonly used methods:

1. `public bool IsMatch(string input):` Indicates whether the regular expression specified in the *Regex* constructor finds a match in a specified input string.
2. `public bool IsMatch(string input, int startat):` Indicates whether the regular expression specified in the *Regex* constructor finds a match in the specified input string, beginning at the specified starting position in the string.
3. `public static bool IsMatch(string input, string pattern):` Indicates whether the specified regular expression finds a match in the specified input string.

4. public MatchCollection Matches(string input): Searches the specified input string for all occurrences of a regular expression.

5. public string Replace(string input, string replacement): In a specified input string, replaces all strings that match a regular expression pattern with a specified replacement string.

6. public string[] Split(string input): Splits an input string into an array of substrings at the positions defined by a regular expression pattern specified in the Regex constructor.

### B. REGULAR EXPRESSION SYNTAX:

Syntax	Description
\W	Match any character that is not alphanumeric
\S	Match any character that is not whitespace
\D	Match any character that is not a digit
\B	Match any character that is not the beginning or end of a word
[^x]	Match any character that is not a digit
[^aeiou]	Match a position that is not one of the characters aeiou
*?	Repeat any number of times, but as few as possible
+?	Repeat one or more times, but as few as possible
??	Repeat zero or one times, but as few as possible
{n,m}?	Repeat at least n, but no more than m times, but as few as possible
{n,}?	Repeat at least n, but as few as possible

Table 1. Regular Expression Syntax [3]

Some Tools or software use to develop the product like Microsoft Visual Studio, SQL Server, Edit Pad pro 7 in this Microsoft Visual Studio is use for the design and for actual coding and for the database and for server use SQL server and last one edit pad pro7 use for developing regular expression and also use for weather RE is correct or not.to view the report we use Microsoft Excel.

### I. SYSTEM ARCHITECTURE

Fig.1 shows the flow of the AMRA.in this firstly upload the university Result Pdf and then convert the pdf in excel format.to generate the student result in detail report format one must have to enter the seat no and get the report. Similarly for

the other report like subject failure student, subject topper student, average report and final report.

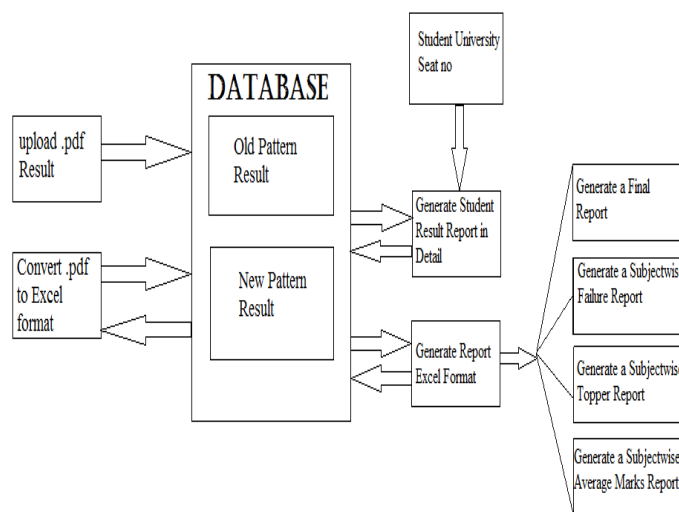


Fig. 1 System Architecture of Automotive Multifaceted Result Analyzer (AMRA)

### II. METHODOLOGY

Algorithm for AMRA is as follows.

- Step 1: Start
- Step 2: Start with Login Window
- Step 3: If Login Successful goto Next Step 5
- Step 4: Else goto Step 2
- Step 5: Upload the Result which is in pdf format
  - Step 5.1: Process the .pdf to .txt format with the help of Regular Expression
  - Step 5.2: View the Result Contents
  - Step 5.3: Save New Pattern Result or Save Old Pattern Result.
  - Step 5.4: Stop
- Step 6: View Result in Report format of student
  - Step 6.1: Result of particular Student Where Input: University Seat No. Output: Detail result of that Student.
    - {Exam Seat No, Student Name, College Code, PRN No, (Result{Paper Code, Paper Name, Paper Type (Theory, Practical, Term Work), Out of Marks, Passing Marks, Out Of 30, Out Of 70, Obtained Marks, Status, Clear Status}}}
  - Step 6.2: Generate the Reports like Top 5 Students, Final Report, Subject wise Failure, Subject wise Topper, and Subject wise Average Marks
    - Input: Type Of Report, academic year, Engineering year, Branch, Paper Name, Paper Type and Top count
    - Output: Report Depends on Type of Report.
- Step 7: Generate Other Reports like
- Step 8: Download the Generated Report in Excel format.

Step 9: Click on Logout Button.  
Step 10: Stop

### III. OUTPUT

#### 1. Login

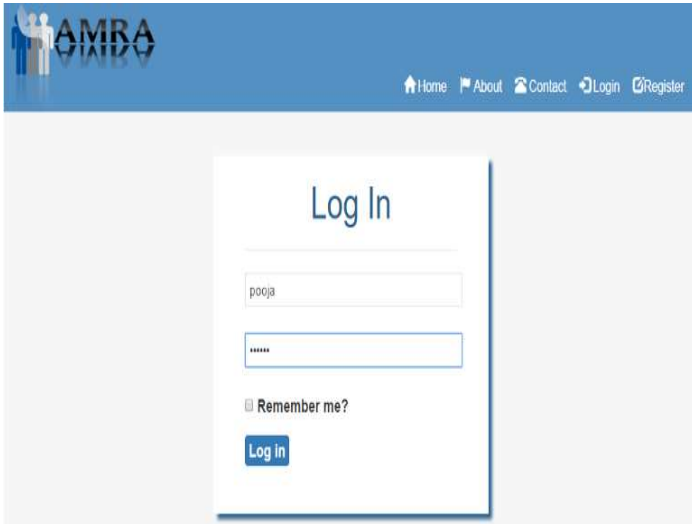


Fig. 2 Login Window

Fig. 2 shows the login window where one must enter valid user name and valid password.

#### 2. Upload PDF

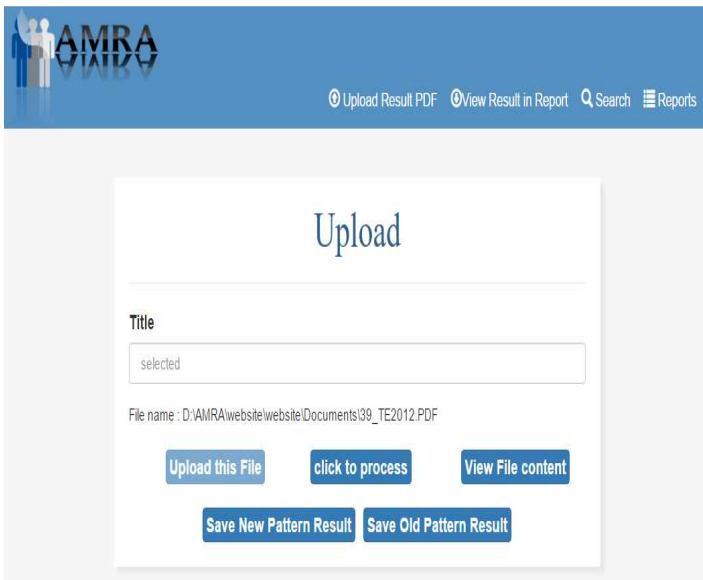


Fig. 3 Upload PDF Window

To convert the result in excel format in system we need to upload pdf result format which is shown in fig. 3

#### 3. View Result

Fig. 4 gives the detail results of pdf format.

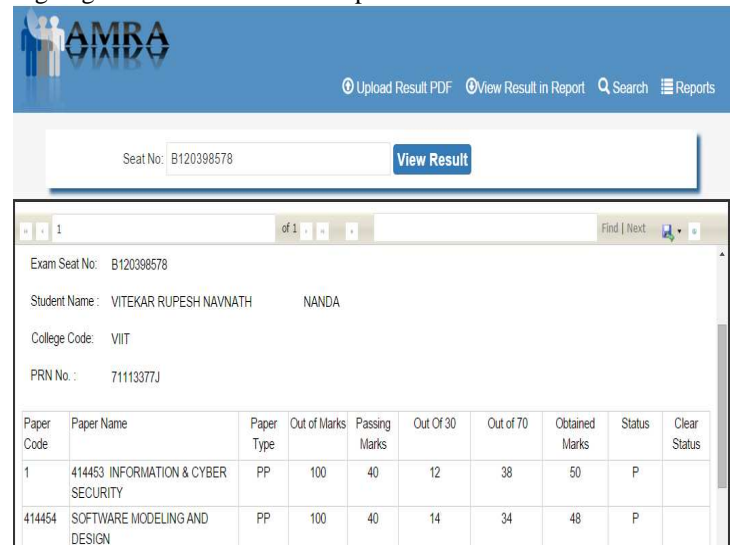


Fig. 4 View Result

#### 4. Result in Excel format

Fig. 5 gives the detail results in excel format once we upload result pdf form.

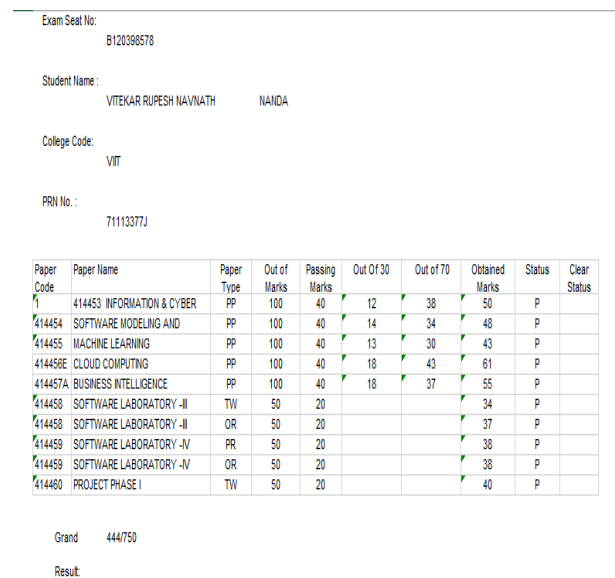


Fig. 5 View Result

#### 5. View Top Five Student

Below fig. 6 shows the report of top five students.

Exam Seat No	Student Full Name	Grand Total	Out Of	Result
T120398570	VALVEKAR AKSHAY VIKAS NEETA	1123	1500	FIRST CLASS WITH DISTINCTION
T120398553	PREETI YADAV DEVI SUKH	1110	1500	FIRST CLASS WITH DISTINCTION
T120398564	SAPTARSHI RUTUJA LAXMIKANT MANISHA	1045	1500	FIRST CLASS WITH DISTINCTION
T120398516	CHANDURKAR MONALI BHASKAR VLIYA	1037	1500	FIRST CLASS WITH DISTINCTION
T120398546	PATIL ASHITOSH PRAKASH KUNDA	1009	1500	FIRST CLASS WITH DISTINCTION

Fig. 6 Result of Top of students

### 6. Top Five Student in Excel Format

Exam Seat No	Student Full Name	Grand Total	Out Of	Result
B80398512	BITAKE YOGESH DATTU TULSABAI	1128	1500	FIRST CLASS WITH DISTINCTION
B80398556	SHELAR YOGITA GAWAJI MANGAL	1125	1500	FIRST CLASS WITH DISTINCTION
B80398528	KANCHAN BHAT KISHINI	1089	1500	FIRST CLASS WITH DISTINCTION
B80398548	PHADATARE PALLAVI POPAT PUSHPA	1083	1500	FIRST CLASS WITH DISTINCTION
B80398523	GOROWADA JASLEENKAUR IQBAL SINGH HARINDERKAUR	1074	1500	FIRST CLASS WITH DISTINCTION

Fig. 6 Result of Top of students in Excel form

Fig. 6 shows the report of top five students in excel form as per our system requirement.

### 7. Inputs for Final Report

Select Type Of Report: Final Report  
 Select academic year: APRIL/MAY 2015  
 Select Engineering year: T.E.  
 Select Branch: INFORMATION TECHNOLOGY  
 Select Paper Name: All  
 Select Paper Type: All

Fig. 7 Input for ISO Format Report

Fig. 7 shows the Input required to generate ISO format report in excel form as per our system requirement.

### 8. Final Report(ISO Report) in Excel Format

Fig. 8 shows the ISO format Reports in excel format

SAVITRIBAI PHULE PUNE UNIVERSITY, INFORMATION TECHNOLOGY (2012 PAT.) EXAMINATION, APRIL/MAY 2015

Paper Name	Paper Type	Distinction	Number of Students with					Appeared	Absent	Passing %
			1st Class	Higher	2nd Class	Pass Class	Fail			
COMP. NW	PP	9	9	13	20	18	0	69	0	100.00%
DBMS	PP	6	10	10	17	25	1	69	0	98.55%
DBMS	OR	24	7	18	12	8	0	69	0	100.00%
DBMS	PR	23	16	11	8	11	0	69	0	100.00%
DES. & ANA. OF	PP	12	5	6	19	23	4	69	0	94.20%
EMP. SKILL DEV.	TW	39	15	9	6	0	0	69	0	100.00%
INFO. TECH PRO	PP	5	6	10	13	34	1	69	0	98.55%
MULTIMED	PP	6	5	8	9	37	4	69	0	94.20%
OPERATING	PP	3	7	15	19	22	3	69	0	95.65%
OS	PR	17	5	5	8	26	8	69	0	88.41%
OS	TW	47	9	5	7	1	0	69	0	100.00%
SEMI. & TEC COM	TW	58	9	2	0	0	0	69	0	100.00%
SOFT	OR	20	5	4	6	24	10	69	0	85.51%
SOFT	PR	25	8	6	8	14	8	69	0	88.41%
SOFT.	PR	22	12	16	8	11	0	69	0	100.00%
SOFT.	TW	42	15	7	2	3	0	69	0	100.00%
SOFTWARE	PP	1	9	13	15	31	0	69	0	100.00%
SYSTEMS	PP	1	6	11	18	32	1	69	0	98.55%
THEO OF	PP	15	9	10	13	22	0	69	0	100.00%
WEB ENGIAND	PP	6	7	12	18	26	0	69	0	100.00%

TOTAL APPEARED	69	Exam Seat No	Student Full Name	Grand	Out Of	Result
TOTAL RESULT	69	T120398570	VALVEKAR AKSHAY VIKAS NEETA	1123	1500	FIRST CLASS WITH

Fig. 8 Result of All students in Excel form

### 9. Inputs for Subject wise Failure Report

Select Type Of Report: Subjectwise Failure  
 Select academic year: MAY 2015  
 Select Engineering year: B.E.  
 Select Branch: INFORMATION TECHNOLOGY  
 Select Paper Name: OBJECT ORIEN. MODELING & DESIGN  
 Select Paper Type: All

Fig. 9 Input for Subject wise Failure Format Report

Fig. 9 shows the Input required to generate Subject wise Failure format report in excel form as per our system requirement.

### 10. Subject wise Failure in Excel Format

Fig. 10 shows the Subject wise Failure Reports in excel form.

OBJECT ORIEN. MODELING & DESIGN ( PP )				Count = 2
Seat No	Name	Marks 30	Marks 70	Total Marks
B80398515	DHANUKA AKSHAY VIVEK SANGEETA			07
B80398541	MAYUR DHAKA ANITA			25

Fig. 10 Result of Subject wise Failure students in Excel form

### 11. Inputs for Subject wise Topper Report

Fig. 11 shows the Input required to generate Subject wise Topper format report in excel form as per our system requirement.

Fig. 11 Input for Subject wise Topper Format Report

### 12. Subject wise Topper in Excel Format

Fig. 12 shows the Subject wise Topper Reports in excel form.

BUSINESS INTELLIGENCE ( PP )		
Exam Seat No	Student Full Name	Obtained Marks
B120398536	KOTHAWADE SHWETA ANIL RATNA	72

CLOUD COMPUTING ( PP )		
Exam Seat No	Student Full Name	Obtained Marks
B120398557	PREETI YADAV SUKH DEVI	72

INFORMATION & CYBER SECURITY ( PP )		
Exam Seat No	Student Full Name	Obtained Marks
B120398524	GORASIA ANKIT DEEPAK ARUNA	74

MACHINE LEARNING ( PP )		
Exam Seat No	Student Full Name	Obtained Marks
B120398575	VALVEKAR AKSHAY VIKAS NEETA	70

Fig. 12 Result of Subject wise Topper students in Excel form

### 13. Inputs for Subject wise Average Marks Report

Fig. 13 shows the Input required to generate Subject wise Average Marks report in excel form as per our system requirement.

Fig. 13 Input for Subject wise Topper Format Report

### 14. Subject wise Average Marks in Excel Format

Fig. 14 shows the Subject wise Average marks Reports in excel form.

Paper Code	Paper Name	Paper Type	Average	Out of Marks
414453	INFORMATION & CYBER SECURITY	PP	53	100
414454	SOFTWARE MODELING AND DESIGN	PP	57	100
414455	MACHINE LEARNING	PP	41	100
414456B	USABILITY ENGINEERING	PP	60	100
414456E	CLOUD COMPUTING	PP	58	100
414457A	BUSINESS INTELLIGENCE	PP	58	100
414458	SOFTWARE LABORATORY -III	OR	38	50
414458	SOFTWARE LABORATORY -III	TW	36	50
414459	SOFTWARE LABORATORY -IV	OR	36	50
414459	SOFTWARE LABORATORY -IV	PR	36	50
414460	PROJECT PHASE I	TW	40	50

Fig. 14 Result of Subject wise Average Marks in Excel form

### CONCLUSION

Colleges under Pune University facing some problem while result analysis like converting the result which is in .pdf format to Microsoft office excel format. Our system gives detail reports such as subject wise report, top student report etc.

Our proposed work is based on following factors like reducing the faculty work and make easier to generate different types reports. Just faculty needs to upload the result pdf and from report panel generate the required report. All reports generated by this work are in the ideal format as per college required format.

#### **REFERENCES**

- [1] Steven A. Smith, “Regular Expressions in Validator ” in ASP.NET by example ,1<sup>st</sup> edition
- [2] Mridula Parihar,” Using SQL Server with ASP.NET” in ASP.NET Bible, Hungry Minds, Inc. 909 Third Avenue New York, NY 10022
- [3] Adrian Turtzchi “Visual Studio.NET IDE” in C# .NET web developer’s, 1st edition
- [4] Steven A. Smith March 2004 , “Regular Expressions in ASP.NET ” [online]Available: <https://msdn.microsoft.com/enus/library/ms972966.aspx>, Aug. 2015
- [5] Steven A. Smith March 2004 , “Regular Expressions in ASP.NET ” [online]Available: <https://msdn.microsoft.com/enus/library/system.text.regularexpressions.regex%28v=vs.110%29.aspx>, Aug. 2015
- [6] Steven A. Smith March 2004 , “Regular Expressions in ASP.NET ” [online]Available: <https://msdn.microsoft.com/enus/library/aa337091.aspx>, Aug. 2015