



# Placify: Seamless Recruitment Experiences Powered by MERN

## *Placify: Seamless Recruitment Experiences Powered by MERN*

*1Prof. V.G.Jagtap, 2Tanishka Shinde, 3Shahid Sayyad, 4Sahil Jagtap, 5Shravan Kale*

*1Professor, 2Student, 3Student, 4Student, 5Student*

*Department of Computer Engineering, SVPM College of Engineering, Malegaon (Bk), Pune, India*

### ABSTRACT

The recruitment process in educational institutions is often inefficient, leading to difficulties for students in securing jobs and for recruiters in finding suitable candidates. Traditional placement methods involve excessive paperwork, manual tracking, and lack of proper communication, resulting in delays and errors. This paper presents the implementation of a Placement Cell Job Portal using the MERN (MongoDB, Express.js, React.js, Node.js) stack, designed to streamline and automate the placement process. The system facilitates real-time job postings, student profile management, recruiter interactions, and Training & Placement (T&P) officer functionalities. It allows students to register, apply for jobs, receive notifications, and track application status. Recruiters can post job openings, filter candidates, and manage hiring processes efficiently. Placement officers can monitor job postings, verify student details, and oversee recruitment activities. By leveraging role-based authentication, the system ensures secure data handling, allowing only authorized users to access relevant information. WebSockets-based notifications provide real-time updates to students regarding job applications and interview schedules. The job application filtering mechanism ensures that students see only relevant job listings based on their qualifications and interests. This implementation aims to reduce manual workload, improve efficiency, and provide an organized structure for job placements. The proposed system enhances data security, automates recruitment workflows, and bridges the communication gap between students, recruiters, and placement officers. The use of MongoDB for data storage, Express.js and Node.js for backend services, and React.js for an interactive UI ensures a seamless and responsive user experience. The project demonstrates how technology-driven recruitment systems can significantly improve the placement process in academic institutions.

keywords: *placement portal, job recruitment, mern stack, training & placement, automated recruitment.*

### I. INTRODUCTION

With the increasing number of students seeking jobs, the traditional placement process has become cumbersome and inefficient. Educational institutions often rely on informal communication channels such as email groups, Telegram or WhatsApp to disseminate placement-related information, leading to data mismanagement, lack of structured communication, and difficulties in tracking student applications [3]. Manual placement management methods frequently result in delays, human errors, and coordination challenges between students, recruiters, and Training & Placement (T&P) officers.

The Placement Cell Job Portal aims to overcome these challenges by introducing a centralized, automated platform that streamlines the recruitment process [1]. Leveraging modern web technologies such as the MERN stack (MongoDB, Express.js, React.js and Node.js) [2], the proposed system offers a real-time, user-friendly, and highly efficient solution to enhance the overall placement experience.

The portal acts as a bridge between students, recruiters, and placement officers, ensuring smooth communication, organized application tracking, and secure data handling. By shifting from manual to digital methods, the system enhances transparency, data security, and operational efficiency, making campus placements more accessible and effective.

#### Objectives

- Develop a centralized job portal to efficiently manage campus placements .
- Provide real-time job updates, application tracking, and recruiter management using modern web technologies .
- Enhance data security and minimize errors in student registration and job applications through secure authentication mechanisms .
- Facilitate seamless communication among students, recruiters, and placement officers with real-time notifications .
- Reduce manual workload and improve placement coordination efficiency by automating administrative tasks .
- Implement a role-based authentication system (using JWT) to ensure secure access control for different users.

### II. RELATED WORK

The author highlights the problem statement how unemployed people struggle to find jobs that fit their skills, while companies have a hard time finding qualified candidates. The Job Portal is created to help solve this issue by offering a web application and an Android app for job searching on both computers and phones. It allows job seekers to register, update their skills, and search for job openings easily. The portal uses Angular and React for the website and Java with Spring Boot for the backend, making it secure and easy to use. While it helps job seekers and employers connect, the process can still take a lot of time and can be stressful, especially for those who aren't very familiar with technology. Overall, the Job Portal aims to improve the job market by matching job seekers with suitable jobs and helping companies find the right talent[13].

The job portal was created to address the competitive job market, providing university students with effective tools to secure placements by bridging the gap between them and potential employers. Existing placement systems are often inefficient, leading to missed opportunities; thus, a centralized platform is essential for easy access to job listings and applications. Key features of the portal include user profiles for managing job applications, detailed job listings with crucial information, and admin access for maintaining current postings. Built using the MERN stack, the portal ensures robust and scalable functionality. While it offers wider reach and centralized information, limitations such as information overload and reliance on internet access may pose challenges for students. Potential technical issues could disrupt access and reduced personal interactions with recruiters might hinder relationship building. Overall, the portal aims to enhance placement rates and streamline the recruitment process for students[14].

Author Name	Technology Used	Advantages	Disadvantages
G Ramarao1 , P Anjanima , E Supriya, Md Abdulrahman, Chakradhar, K Karthik	Angular, React, Java, Spring Boot	- Easy job search for users - Web and mobile access - Cost and Time efficient	- Time-consuming matching process - Stressful for tech-averse users
Niraj Srivastava, Manashvi Tripathi, Dr. Vipin Rai	MERN Stack	- Centralized access to job listings - User profiles for managing applications	- Information overload
Shubham B. Gulik, Akash R. Gharat, Jayesh L. Choudhary, Sujata Kolhe	Python (Beautiful Soup), SMTP	- Email notifications - Job-saving options	Complexities of web scraping - Scalability concerns
Dr. N.Swapna Goud, B. Siva Krounchidhar Sarma, B. A shrithNivyadin, C. Shasank	MERN Stack	- User-friendly interface - Advanced job search options	- Privacy concerns - Inaccurate job matches
Dr. Gauri Dhopavkar, Toshit Kale, Rajat Gaikwad, Saurabh Chavan, Gaurang Kumar, Himanshu Shendare, Harsh Akre	MEAN Stack	- Automation of manual processes - Improved communication	- Lacks performance tracker for mock tests - Manual filtering required
Aishwarya Baile, Pratik Sawarkar, Pratik Wankhede, Rajat Manwatkar, Raman Barsagade, Prof. Preeti Karmore	Machine Learning	- Automates recruitment tasks - Improved efficiency and data accuracy	- Excludes less tech-savvy students - Significant investment required

Varun Shenoy & P. S. Aithal	Python, SQL	- Simplifies scheduling - Enhanced job placements	- Reliance on technology for operation
-----------------------------	-------------	--	--

Fig 1 :Literature Survey Table

The demand for a quicker and more streamlined job search experience has inspired the creation of a job portal that uses web scraping to gather listings from different sources, helping users save both time and effort. The portal includes features such as email notifications, job-saving options, and the potential for automated interview scheduling, enhancing the user experience. By using BeautifulSoup in Python for web scraping and SMTP for email notifications, the platform successfully centralizes job opportunities. However, challenges include understanding the complexities of web scraping, scalability concerns, and reliance on external job portals, which may change over time. Despite some challenges, the platform offers valuable benefits by making the job search easier and giving users quick, centralized access to job listings[15].

Job portals aim to make job searching easier and recruitment faster by connecting job seekers with employers. They are designed to be user-friendly, offering advanced search options for filtering jobs based on criteria like location and skills. However, there is a lack of proper studies on how satisfied users are with these platforms. The study found users were generally happy with the variety of jobs and features offered, but also identified issues like privacy concerns and inaccurate job matches. Built using the MERN stack, the portal shows potential, though improvements are needed to address data privacy and better match users with relevant jobs[16].

The motivation behind developing the Training & Placement (T&P) portal is to address challenges in traditional placement systems, such as manual processes and poor communication, which make it difficult for students to secure jobs efficiently. The problem lies in the inefficiency of current manual systems that cause delays and errors in managing student placements. The system automates tasks like student registration, job notifications, and communication, while also analyzing performance data to predict suitable job opportunities. Built using the MEAN stack, the platform is scalable and effective, although it lacks a performance tracker for mock tests and requires manual filtering for certain tasks. The results show that the T&P portal has improved efficiency and communication between students and placement officers, though it relies heavily on technology and may have an initial learning curve for users[17].

The Online Campus Recruitment System simplifies the recruitment process by automating tasks like job postings, interview scheduling, and results management, making it easier for students and companies. It uses machine learning to evaluate candidates and allows students to access tests online. This improves efficiency and data accuracy but has some limitations, like excluding less tech-savvy students and requiring significant investment. It may also overlook qualified candidates who don't perform well in tests, but overall, it enhances communication and reduces manual work[18].

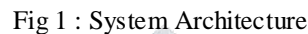
This project seeks to make the recruitment process smoother and more efficient for college students and companies by moving away from traditional methods that can be slow and often lead to mistakes. The system includes features like separate logins for students, companies, and admins, along with profile management and admin-controlled data accuracy. It also allows for pre-interaction between students and companies, reducing interview anxiety. Developed using Python and SQL, it simplifies scheduling and enhances job placements. However, users may face reliance of technology may cause disruptions during technical failures or system downtimes. Despite this, it offers improved efficiency, data security, and better opportunities for students[19].

The online campus placement system improves efficiency and accessibility for both students and recruiters. It allows for electronic submission of CVs, online screening, and remote interviews. The study identified 158 key factors that influence the model, highlighting increased flexibility and reduced logistical needs. However, the system also faces challenges like technical issues, high costs, and students being less prepared for traditional interviews. Overall, while it offers many advantages, some drawbacks, such as system errors and added stress, remain significant[20].

The study highlights how online job portals have made job searching faster and more efficient, especially in a competitive market. They offer features like quick job applications, filters, and resume builders. However, issues like too many job listings and poor quality posts can affect job seekers. Employees hired through these portals perform well, but there are concerns like fake job listings and a lack of representation for certain roles. Despite these challenges, the portals increase access to opportunities and speed up the hiring process[21].

### III. SYSTEM ARCHITECTURE





For example, some job portals rely on third-party APIs for data aggregation, resulting in inconsistent updates and unreliable job postings. Others use generic resume parsing algorithms that fail to match candidates effectively based on their skills and job preferences.

- **Secure Authentication:** Implements JWT-based authentication to ensure only verified users can access job postings and applications.
- **Real-Time Notifications:** Uses WebSockets to provide instant updates about job postings, interview schedules, and application status.
- **Smart Job Filtering:** Implements custom job recommendation algorithms based on student profiles, recruiter preferences, and job market trends.
- **Alumni & Career Insights:** Maintains records of previous successful placements to help students learn from past hiring patterns and better prepare for interviews.

#### IV. TECHNOLOGY USED

### Key Technological Components:

1. MongoDB (Database Layer)  
Type: NoSQL, Document-oriented database.  
Stores data in a flexible JSON-like format, allowing for dynamic schema updates.  
Efficiently handles large volumes of unstructured data, such as student profiles and job listings.  
Supports horizontal scaling, ensuring the system can grow with increased users.
2. Express.js (Backend Framework)  
Type: Lightweight web application framework for Node.js.  
Simplifies server-side logic and API creation.  
Provides middleware functions for handling requests, authentication, and database interactions.  
Enhances security and performance with built-in HTTP utilities and error handling mechanisms.
3. React.js (Frontend Library)  
Type: JavaScript library for building user interfaces.  
Utilizes a component-based architecture, making UI development modular and reusable.  
Enables fast rendering with a virtual DOM, improving user experience.  
Supports state management and dynamic content updates, ensuring real-time job postings and application tracking.
4. Node.js (Runtime Environment)  
Type: JavaScript runtime built on Chrome's V8 engine.  
Enables server-side execution of JavaScript, unifying the technology stack.  
Uses a non-blocking, event-driven architecture, making it highly efficient for handling multiple user requests.  
Provides built-in asynchronous support, improving real-time performance for chat, notifications, and job alerts.

By implementing role-based access, secure authentication and real-time notifications, the system optimizes placement processes for students, recruiters, and placement officers.

#### 4.1 Algorithm : Smart Job Filtering (Pseudo Code)

This algorithm filters job listings based on student profile attributes such as CGPA, branch, skills, and graduation year.

Input: StudentProfile, JobList

Output: FilteredJobs

1. Initialize FilteredJobs as an empty list
2. For each job in JobList:
  - (a) If StudentProfile.cgpa  $\geq$  job.min cgpa
  - (b) AND StudentProfile.branch  $\in$  job.eligible branches
  - (c) AND intersection(StudentProfile.skills, job.required skills)  $\geq$  threshold: Add job to FilteredJobs
3. Return FilteredJobs

#### 4.2 Algorithm : Real-Time Notification (Pseudo Code)

This algorithm uses WebSocket to send real-time updates to students when job statuses change.

Input: event (job posted / application status updated), user id

Output: Notification delivered to the connected user

1. Open WebSocket connection on client and server
2. On event trigger:
  - (a) Identify target user id or group
  - (b) Construct message payload
  - (c) Send message via WebSocket
3. Client receives message and displays notification popup

## V. RESULT ANALYSIS

The Placement Cell Job Portal has significantly enhanced the efficiency of campus recruitment by automating job postings, application tracking, and communication between students, recruiters, and placement officers. The analysis of its performance highlights the following key outcomes:



Fig 5.1: Home Page

The Home Page welcomes users to the Placement Cell portal, highlighting its mission to connect students with leading companies and support their career growth. It emphasizes placement training and industry collaborations that provide valuable opportunities. The page serves as an introduction to the platform's features and benefits for students preparing for successful careers.

Fig 5.2 : Student Registration Page

Students can register by filling in their name, email, password, role and contact details. This form supports profile picture uploads and role-based registration (Student/TPO/Recruiter). It is the entry point to access placement-related features. Registered users can securely log in to explore job opportunities and placement tools.

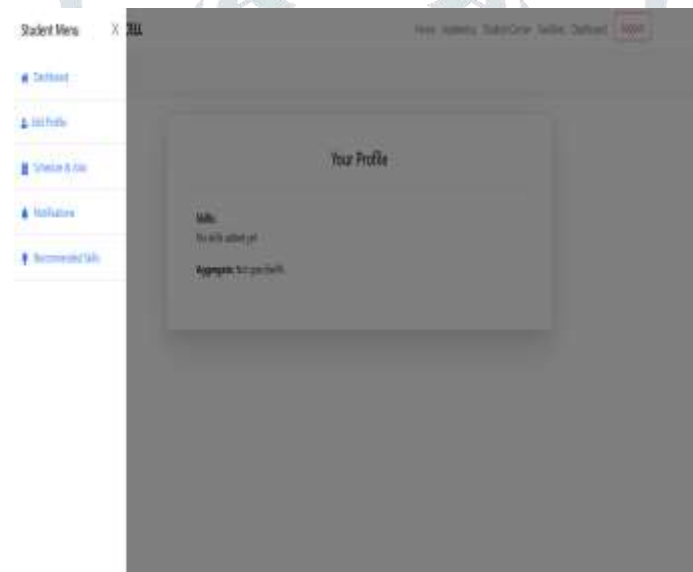


Fig 5.3 : Student Dashboard

The student dashboard provides access to edit profile, view jobs and get skill recommendations. It shows the student's current skills and aggregate percentage (if updated). This panel helps students manage their placement readiness and receive relevant updates.

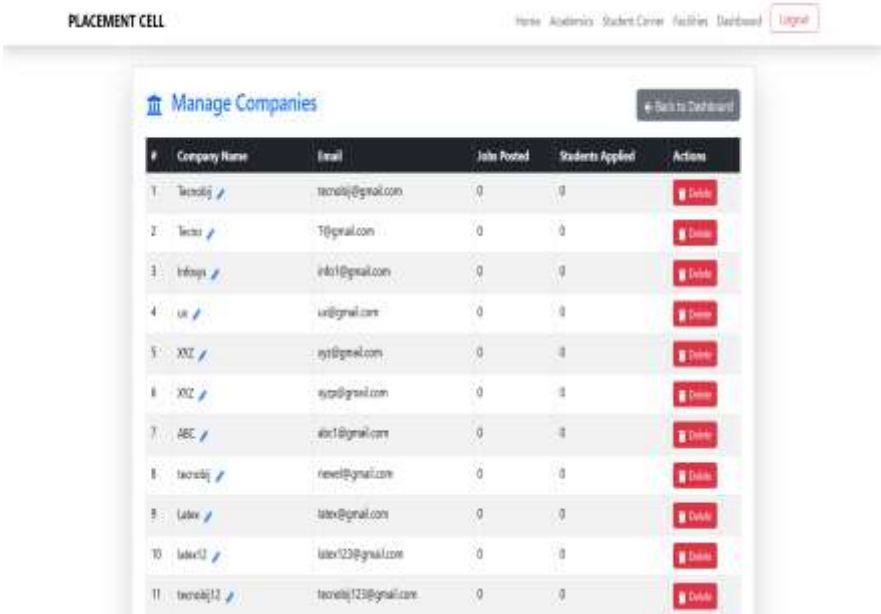


Fig 5.4 : Manage Companies Page

The admin can view, edit, and delete company entries from a structured table. Details such as company name, email, jobs posted, and students applied are shown. This helps manage and monitor TPO data efficiently.

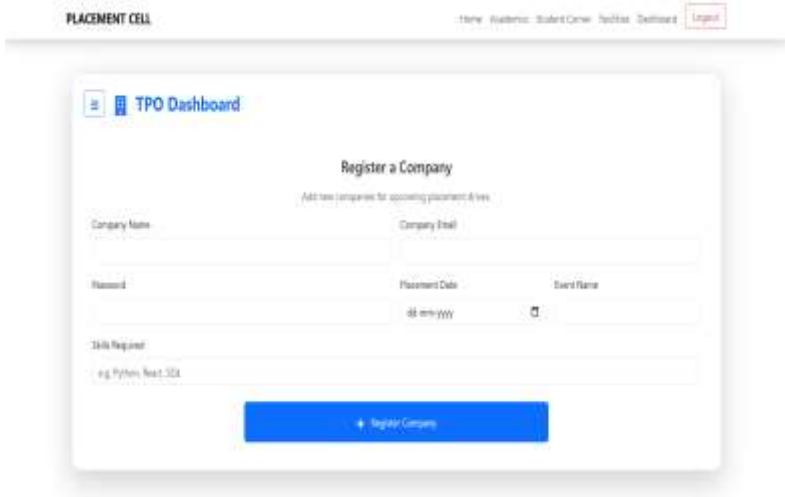


Fig 5.5: TPO Dashboard

TPO (Training and Placement Officer) can register companies for placement drives. Fields include company name, email, required skills, date and event name. This feature streamlines planning and scheduling of recruitment events.

VI. FUTURE SCOPE

The Placement Cell Job Portal has the potential to evolve further by integrating advanced technologies to enhance user experience and recruitment efficiency. Future improvements may include:

- **AI-Based Job Recommendations:** Implementing machine learning algorithms to analyze student profiles, skills, and job preferences to provide personalized job recommendations. This will improve job visibility and help students find suitable roles more efficiently.
- **Mobile App Integration:** Developing a mobile-friendly application to improve accessibility for students, recruiters, and placement officers. This will allow users to apply for jobs, receive notifications, and track applications on the go.
- **Automated Resume Screening:** Leveraging AI-driven resume evaluation tools to analyze resumes based on keywords, experience, and qualifications, allowing recruiters to filter and shortlist candidates quickly.
- **Virtual Interview Scheduling:** Integrating a built-in scheduling system where recruiters can set up and manage video interviews directly through the portal, reducing dependency on third-party services.
- **Real-Time Career Insights:** Introducing data analytics dashboards that provide insights into placement trends, recruiter engagement, and job market demands, helping institutions tailor training programs to industry needs.
- **Blockchain-Based Credential Verification:** Implementing blockchain technology to verify student qualifications and certifications, reducing fraud and ensuring the authenticity of candidate profiles.

- Integration with Industry Partners: Expanding collaborations with companies by providing API integrations with corporate hiring platforms, ensuring seamless recruitment pipeline management.

By incorporating these advancements, the Placement Cell Job Portal will continue to enhance job search efficiency, recruitment automation, and overall user engagement.

## VII. CONCLUSION

This paper presents the implementation of a Placement Cell Job Portal using the MERN stack, addressing the inefficiencies of traditional placement systems. The proposed solution provides real-time job updates, recruiter-student interactions, and automated placement tracking, significantly improving the efficiency and transparency of campus recruitment.

By leveraging MongoDB, Express.js, React.js, and Node.js, the system ensures scalability, secure authentication, and structured job management. Students benefit from an intuitive job search experience, recruiters gain access to a well-filtered candidate pool, and placement officers can monitor and streamline the hiring process effectively.

The portal's role-based authentication, real-time notifications, and job filtering algorithms enable faster hiring processes, improved candidate-job matching, and enhanced data security. Additionally, automation minimizes manual intervention, reducing administrative workload for placement cells.

Future enhancements, such as AI-driven job recommendations, mobile app integration, and automated resume screening, will further improve the system's effectiveness. The successful implementation of this portal demonstrates how technology-driven recruitment platforms can revolutionize campus placements, creating a more structured, accessible, and efficient hiring process for all stakeholders.

## VIII. REFERENCES

- [1] M. Smith, "Automated Job Portals: Enhancing Recruitment Processes," *International Journal of Computer Science*, 2024.
- [2] N. Kumar, "MERN Stack for Web Applications," *Springer*, 2023.
- [3] A. Gupta, "Campus Placement Management Systems: A Review," *IEEE Conference on EdTech*, 2022.
- [4] R. Williams, "The Role of AI in Recruitment Automation," *Journal of Artificial Intelligence and Data Science*, 2021.
- [5] T. Lee, "Secure Authentication Mechanisms in Web Applications," *International Journal of Cybersecurity and Privacy*, 2023.
- [6] P. Sharma, "Blockchain-Based Credential Verification in Recruitment," *Journal of Digital Innovations*, 2022.
- [7] J. Brown, "The Impact of Cloud Computing on Scalable Job Portals," *ACM Computing Surveys*, 2023.
- [8] S. Patel, "Real-Time Notifications Using WebSockets in Recruitment Systems," *Journal of Web Technologies*, 2021.
- [9] L. Garcia, "Improving Job Recommendation Algorithms with Machine Learning," *Journal of Data Science and AI*, 2022.
- [10] H. Kim, "Mobile App Integration for Recruitment Platforms," *International Conference on Mobile Computing*, 2023.
- [11] B. Singh, "Comparative Study of Job Portals Based on Different Web Technologies," *International Journal of Software Engineering*, 2024.
- [12] K. Das, "Enhancing User Experience in Job Portals with React.js," *Frontend Development Journal*, 2022.
- [13] G Ramarao1 , P Anjanima2 , E Supriya3 , Md Abdulrahman4 , Ch Chakradhar5 , K Karthik6," Job Portal", International Journal of Advanced Research in Science, Communication and Technology (IJARSCT) Volume 4, Issue 2, May 2024.
- [14] Niraj Srivastava, Manashvi Tripathi, Dr. Vipin Rai, "The Development of a Job Portal to Facilitate In-campus Placement", 2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N) ISBN: 979-8-3503-3086-1/23/\$31.00©2023 IEEE 1549.
- [15] Shubham B. Gulik, Akash R. Gharat, Jayesh L. Choudhary, Sujata Kolhe," Scraping of Job Portal", International Journal of Advance Research, Ideas and Innovations in Technology, ISSN: 2454-132X Impact Factor: 6.078 (Volume 7, Issue 3 - V7I3-1748).
- [16] Dr. N. Swapna Goud, B. Siva Krounchidhar Sarma, B. Ashrith Niviyadin, C. Shasank," Job Portal: MERN Stack Web Application for Job Seekers and Recruiters", International Journal of Multidisciplinary Research in Science, Engineering, Technology & Management (IJMRSETM),ISSN: 2395-7639,Volume 11, Issue 4, April 2024 .
- [17] Dr. Gauri Dhopavkar, Toshit Kale, Rajat Gaikwad, Saurabh Chavan, Gaurang Kumar, Himanshu Shendare, Harsh Akre," An Integrated Web Application for Training and Placement", ISSN: 2094-0343 2326-9865.
- [18] Dr. Mithilesh Kumar Dubey , Barkha Narang," Online Campus Recruitment System-A Machine Learning Model", International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 11 Issue: 10s.
- [19] Aishwarya Baile, Pratik Sawarkar, Pratik Wankhede, Rajat Manwatkar, Raman Barsagade, Prof. Preeti Karmore," Implementation of Training and Placement System", International Journal of Scientific Research in Science and Technology Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijrsst.com) doi : <https://doi.org/10.32628/IJSRST>.
- [20] Varun Shenoy & P. S. Aithal," ABCD Analysis of On-line Campus Placement Model", IRA-International Journal of Management & Social Sciences ISSN 2455-2267; Vol.05, Issue 02 (2016) Pg. no. 227-244 Institute of Research Advances <http://research-advances.org/index.php/RAJMSS>.
- [21] Ahjaz Ahmed," A Review on the Impact of Online Job Portals on Employee Performance", International Journal of Research, e-ISSN: 2348-6848 p-ISSN: 2348-795X Vol. 10 Issue 08.