Internship Portal

- Database Project Report

Team Name : Radiance

Shachi Deshpande	-	140110047
Deep Modh	-	140050002
Ashna Gaur	-	140050087
Neeladrishekhar Kanjilal	-	140050007

Index

Overview

The Internship Portal is used heavily during the Internships Season at IITB, however we need a lot of improvements in it. Improvements are needed in both the graphical interface and the kinds of queries possible. Many essential functionalities can be added. Verification of resumes submitted by students is an important activity, and has to be completed before commencement of Internships season, and right now there is no interface available in the portal to automate the verification process. Hence this process of verifying different points mentioned in resumes is a tedious manual process where students have to separately mail the proofs and get resumes verified. We plan to automate and streamline this process, so that both the verifying authorities and students are benefitted.

Also, when students sign for different companies, we would like to display some informative facts and reviews about the companies by previously selected students for the respective companies. We plan to add this feature to the portal, and hope that this will help the students considerably in the decision process of signing for the company. Thus we have chosen to design and implement the database for this portal.

Goals

1. We are planning to design Internship Portal for IITB, so that the current portal is improvised. We plan to insert features so as to get all different kinds of queries possible.
2. Introduce a new module in the existing design to automate the process of resume verification
3. Introduce a new module to collect the reviews of different companies from the finally shortlisted students after completion of their respective internships and meaningfully display the trends in reviews to the future batches of students who might want to apply for the same company
4. Also, we should be able to generate statistics for different batches, companies, etc., which will be helpful in various managerial aspects of the internship recruitment process, like scheduling order of companies.

Functionality and Features Implemented

Basic functionalities:

1. Students register themselves to the Internship Portal and add their personal details. The details like CPI are ideally extracted with help of ASC but for demo purpose we would use dummy CPI data. Thus these records get added to corresponding student Relations.
2. Companies register themselves to the portal similarly and their data gets added to the Company Relation
3. Company is able to add different IAF's according to the job profiles they offer and these IAF's get added to the Relation 'iaf'.
4. Students upload resume, SOP and optionally a profile photo which gets added to the Relation 'Files'.
5. In the time window of verification, students upload different pdf/images for certificates and give LDAP ID's of verifying authorities for points in resume which require verification by faculty or some other people, as described in detail ahead..
6. After this coordinators (who are students with special admin rights) verify the documents uploaded and check if all the verifying authorities have indeed validated the corresponding points in resume. If everything is good, this resume gets 'activated' and can be used to sign for companies
7. Different IAF's are opened by the coordinators on the opens_on date of the IAF and Student can view the IAF and sign it with one of the several resumes/SOP he has submitted. Once the IAF closes, the students who signed the IAF by default get shortlisted for round '0' of selection and hence form a part of the shortlisted students Relation with status set to 0 everywhere.
8. Student can view the all the IAF's (s)he has signed and their status (whether the student is shortlisted for the current selection round in progress).
9. The companies can see the list of students who have signed up the IAF, and the primary personal student details like CPI, department etc. Companies can shortlist students via portal at various rounds.
10. Admin (coordinators) can view, edit, open and close the IAF. She/he can view the results comprising of different shortlists , download the resumes of candidates, download the csv with key personal details.

Special Additions: (features not in current IITB internship portal)

Resume verification: We aim to develop a portal where the student can get his/her resume verified by the concerned authority by specifying the Ldap of the authority(if it exists).
We had planned to create a system where the concerned person can login using their ldap and password to see the resume of the student, point to be verified and get a ‘verify’ button for verifying and validating the specific point mentioned in resume. However we have not added this feature currently due to lack of time. But we plan to add this feature to streamline the process of verification.

The final verification authority though remains with the admin (internship coordinator) and the resume can be activated only when the final verification is done.
Also the student can upload images/pdf’s for soft-copy proofs, which are generally certificates, mark-sheets etc.

Now the coordinators should be able to check if the provided proofs are sufficient and then mark the resume as ‘verified’ after which student is able to sign for companies using this resume.

Review system: A student who does an internship for a particular company/university can submit the feedback for the internship after it is over. This review portal would be open only for a certain period of time after the internship season ends. This timeline is decided by the admin(the head of internships). This review can be viewed by students of next batch while they sign for that company for internship.

How to Setup and Run the System

Pre-requisites :

1) We need mysql2 (version between 0.3 to 0.5)
2) Ruby 2.3 or higher is to be installed on the Linux system.
3) Run command '***gem install rails***'.
4) Running the command '***rails --version***' should now show a version of at least 5.0.0.

Setting-up :

5) Go to terminal and open mysql.
6) Run following commands.
   create database company_development;
   grant all privileges on company_development. to 'companyAdmin'@'localhost' identified by 'password';*

7) Now extract the project folder 'company' and open it in terminal
8) Now run command '***bundle install***'
9) Run the command '***rake db:migrate***'
10) Run command **'mysql -u root -p company_development < script.sql**'

Running Project :

11) Run command '***rails server***'. Rail server has url http://localhost:3000/ which is Login screen.
12) Sign up student and you are good to go.

Explanation of Interface :

a) http://localhost:3000/

This login and password can be generated via the 'Sign Up' option where a student registers himself/herself. We are then able to log in and see the home page for the student. We see a summary of his placement details. We see an index of IAF's he has signed so far, their status, etc. once his resume is uploaded and then verified by the admin. We can edit the personal details of the student, allow the student to 'sign' or 'unsign' the IAF, and show his results for each IAF. When an IAF is signed, this student entry is added to shortlisted students table so that company is able to see the name of student under the corresponding IAF.

The student can upload his resume using the home tab and then click on 'upload resume'. In the resumes tab the student is able to add points to be verified using 'Add point' button.

The status of these points is shown as 'pending' beside each point until the points are reviewed and verified by the admin.

When the student is selected in a company, (s)he would see the status as 'Selected for the company XYZ' and his/her name would be removed from the shortlist relation for all other IAFs.

b) http://localhost:3000/

Here, we similarly see the home page of the company after clicking on the button company login. (The login for company can be implemented similar to student login but hasn't been implemented here)
We see a list of IAF's the company has added. The company can add new IAF's, view the existing IAF's and edit them as well until freezed. The company can also see the list of students who have 'signed' for the company by clicking on IAF and then Signed students link inside the IAF. The company can shortlist the students by a link 'Shortlist' after every student entry. This basically modifies status attribute value in a tuple to the relation 'shortlists'. Hence once the company clicks on 'shortlist' against the names of few students, and then clicks on 'Freeze shortlist' then the result of the IAF gets frozen. Thus no students will be able to see this IAF in their home or IAF tab.
Also, the tuples corresponding to this student which were added to shortlists table will get removed from the signed students list from all other IAFs to prevent the selection of same student in 2 or more places.

Setting up admin :

c) From the same link http://localhost:3000/

we can log in as an admin if admin privilege is added to that particular user (i.e. student) via mysql command line (Make the value of admin field of that student equal to 1)
So to create admin, we first need to sign up as student. After this, we can run sql command which changes the 'admin' attribute of this student as described above.
Now if the admin logs in like a student, he/she is able to get all the admin privileges.
Query : ***'UPDATE students SET admin = true WHERE id = id_of_admin;***'

We can see all the IAF's added so far to the system, the list of companies and students registered, etc. S(he) is able to add/view/edit entries in any of the relations. The paths are much similar to the paths in company/student login.
More importantly we can verify the resumes of students who sign up and upload their resumes. Until the admin verifies the student resume's, the student won't be able to see/sign IAF's using that resume.

Database Design

Relations and their attributes in Database:-

1) **Student Related Relations**: Following sets of attributes are related to student.
    a) **Student Personal Details:**
        1. *ID* (primary key)
        2. *Name*
        3. *Roll Number*
        4. *Department*
        5. *Batch*
        6. *Program*
        7. *Category*
        8. *Dob*
        9. *Nationality*
        10. *Sex*
        11. *Gpo-id*
        12. *Alt_email*
        13. *Hostel*
        14. *Mobile*
        15. *Alt_mobile*
        16. *Home_contact*
        17. *Permanent_addr*
        18. *Specialization*
        19. *Skype*
        20. *email*
        21. *Password_hash*
        22. *password_salt*
        23. *verified*
    b) **Student Placement Details**:
        1. *Year*
        2. *Semester*
        3. Is_job_assigned
    c) **Files**:

        Resume (some attributes related to content of resume file included)

2) **Company**

    d)
        1. *ID* (unique ID for the company as primary key)
        2. *Name*
        3. *Address*
        4. *City*
        5. *Country*

6. *Pin*
7. *Url*
8. *Contact_person*
9. *Designation*
10. *Email*
11. *Contact_number*
12. *Nature*

e) This Relation stores details for a company participating in the internship recruitment.

3) **Iaf (Internship Announcement Form)**

a) Attributes-

1. *IAF_id*
2. *ID* (from Relation Company)
3. *Profile* (could be Finance, Software etc.)
4. *Place_of_posting*
5. *Job_description*(String to hold specific job description)
6. *Stipend*
7. *CPI cutoff*
8. *Selection_process* (string to hold the selection process details of company)
9. *opens_on* (Date and time when IAF is opened for students)
10. *closes_on* (Date and time when IAF closes for students)
11. *Departments cse, elec, mech, meta* (These are boolean attributes indicating whether the IAF is open for that particular department)
12. *Status* (Indicates if IAF is active)

b) This is weak entity dependent on Company. Every company basically adds one or more IAF's for the different job roles that they are offering.

4) **Shortlists (List of shortlisted candidates)**

a) Attributes-

1. *Student_id* (unique ID identifying the student who signed IAF)
2. *IAF_id* (unique ID for an IAF floated by company)
3. *Status* (Selection status of the student)
4. *Created_at* (Day when student with Student_id signed an IAF with IAF_id)
5. *Updated_at* (Day when the status of selection of a student is changed)

b) This is weak entity dependent on 'Student Personal Detail' and 'Internship_application_form'. Thus attributes *ID* and *IAF_id* from the respective strong entities form primary key for this Relation

5) **Coordinator**

a) This Relation holds the details of students who are themselves Internship Coordinators managing the companies and hence the recruitment process.
b) Attribute *'admin'* in student is set to '1' to make him/her a coordinator.

6) **Verification**

a) Attributes-
   1. *ID*
   2. *Point_to_verify* (String describing what is to be verified)
   3. *Verifying_authority* (LDAP of authority who verifies the point)
   4. *Verifying_document* (pointer to document which might be a mark-list or certificate image/pdf)

b) For a particular point on resume, either of the two attributes- *Verifying_authority* or *verifying_document* need to be filled up according to the verification guidelines.

7) **Review**

a) This relationship exists between company and shortlisted students. Only the finally shortlisted students are able to write a review for the company after their internship period is over.
b) A string *Description* and *Timestamp* of when the review was given are the two attributes for this relation.
c) Thus the attributes are-
   1. *ID*
   2. *Organisation*
   3. *Description*
   4. *Time_of_review*
   5. *Student_id*

8) Shortlist

This relation holds information about students shortlisted for any particular IAF.

a) *IAF_id*
b) *Student_id*
c) *Status*
d) *Created_at*
e) *Updated_at*

Future Work

1) We would have liked to create a login system for the professors so that the professors see the list of students who have requested verification on their home page. The Professor would be able to see the student details upon clicking on the student name, along with the points in resume to be verified. Now the professor can just verify the point with one click of button and then this point in the relation ‘verification’ for this student will be marked as verified. This will streamline the verification process to a large extent.
2) We would like to add a statistics tab to this app for admin which can show certain statistics like number of students selected by each company registered (both graphically and as a list), CPI-wise trends of students getting selected for internships, number of students selected for internships in each department under various categories, and so on.
3) We could also add some special tabs on the home page of the coordinator (who basically has admin rights) so that (s)he is able to maintain a track of what companies (S)he is handling and what is the status of selection process for each of the companies.

Future Scope

This web app can be used to streamline the internships season at IITB. We currently have an app to handle this, but it does not have the functionalities related to managing the resume verification process, which is a really tedious and crucial process to be completed before a student can sign an IAF. The Internship Coordinators have to ask students to mail the proofs supporting various points in the resume and manually go through the proofs. This becomes process becomes much easier in this web app. After adding a login system for the professors, the work of the professors and other verifying authorities would also reduce to a large extent.

The Internship Coordinators have to currently maintain several data sheets to track the status and statistics of developments in internship season. This could be well-managed after addition of a statistics module to the current project.

Currently students have to contact several senior students to find about experience of internships at specific companies. Though this process is fairly efficient, we have also added a review system to our app where students can add a review of the company at which they get selected. Thus the future batches would be able to see the reviews of their seniors directly after they log in to this app. We can improve this system much more by associating the list of reviews to the specific company name, and also adding a system where students can rate various aspects of work experience. The future batches of students would be able to see this statistics associated with each company as well.

We can create a notifications system for the web app such as the students are notified via email/message at the end of each day which summarises the IAF's added, and also notifies the students when they are shortlisted/finally selected. This will largely reduce the load of the Coordinators to separately notify the students regarding such developments.
The Coordinators should be able to compose a mail in this web-app such that they are able to also generate customised notifications for any group of shortlists regarding the interviews/group discussion time-slots, etc.

Thus this web-app can be extended in a meaningful way and can be practically used for the management of internship season which affects more than 1600 undergraduate students at our institute (including second and third year students).

Experience

We have worked with Django, Java and Ruby-on-Rails uptil now. Hence we felt that we could write our individual experiences regarding each of these frameworks.

Shachi: I liked Ruby-on-Rails because of the ease of development. We can define methods in controllers and easily produce the desired actions corresponding to common database operations like adding new tuples, deleting, updating etc. We get great templates to work with and hence it is easier to create awesome UI. However Java gives a finer control and independence of back-end development with front-end development. Hence I feel that it is better to work with Java when we want to develop back-end modules independent of front-end.

Ashna: I had previously worked a bit on Ruby on Rails during my summer internship. I am particularly not a fan of Java, and Ruby on Rails therefore was something handy for me to work with. Ruby most of the times provide framework on how to proceed about things. Its very easy to figure out the paths, establish relationships between models and a very easy and comprehensive documentation for reference. In generating views, the variables are carried forward and can be directly referenced in view files giving the developer a full knowledge of how things are going about.

Neeladrishekhar: I had previously worked on Java, Android, Python(Django) and used Ruby(Ruby on Rails) in this project, of which I personally liked Java Servlets for backend and JSP for frontend separately. The main reason for that being the convenience of developing code independently for different ends, which in case of an MVC model (Django, RoR) we have to do all the three sides in parallel and hence extending a feature leads to changes in all ends. The convenience in parallel development is lost hence Java being still my favourite. Because of the difficulty in merging a pure html, css front end to RoR views, java's independent servlets are direct to integrate.

Deep : I have worked with django in python, jdbc-servlet in java and ruby on rails in this project. If I were to do this project again, I will do it in ruby on rails because it is easier to work with after 'steep' learning curve. In RoR we can easily handle database, views and actions. It uses Model-View-Controller. Django in python is similar but API in rails is more coding-friendly. In JDBC front-end and back-end can be implemented separately however in rails you need to consider both of them simultaneously. Easier work interface in RoR overcomes this drawback.

Post-project Demo

Changes : (As suggested by Prof)

1. Avoid race conditions in freezing: We were using a temporary table for certain updates to database and it was pointed out that concurrent invocation of that method would create problems. Hence now we are using temporary tables to generate intermediate tables

For future :

2. Prof login: As already stated, we would like to create an independent login for the professor, where he/she is able to see a list of students who have asked for verification, The professor would be able to see the list of students who took a certain course together. The resume points to be verified would appear adjacent to his/her name.So the professor won''t have to see a cluttered mailbox!
3. Prepared statement: We would like to include prepared statements to prevent sql injection attacks in all the sql queries used within the app