Project HireUp interview notes:

* Start explaining the project with the problem statement.
* Tools, Technologies, and platform used
* Explain the modules and its process of how you developed the solution of module.
* Explain the obstacles you faced in the process and how you solved them.
* Specify about the skills and features needed to complete the project
* Have a positive mindset
* Discuss the goals of the project
* How interesting the project was
* What made you accept this project
* Be a little careful
* Do not get in the detailed project information
* Never criticize your previous employers
* Do not drag your answer
* Your role in the project.
* Your imagination or creativeness in the concept of the project.
* Brief them the mistakes to avoid

Advantage points:

1. Reduce code length for database connection

HireUp – Modules:

1. Registration and Login (Login, singup, forget and reset password)  
   Student – Registration and login  
   Admin – Only Login
2. User Information (User add personal details, academic details, skills details)
3. User settings module
4. User view resume
5. Admin Home page
6. Admin view all students -> mark as selected concept
7. Admin add student hired-up data
8. Admin view selected student
9. Admin view contacts messages
10. Error Page for admin and its concept like admin login error or other kind of error
11. Error Page for user, its concept like user login error or other kind of error

Good Afternoon sir,

My name is Ashish Agrawal and my project partner is Nikhil jain, we together have created an application called as HireUp – The placement management application.

**Problem Statement / ABSTRACT:**

* Managing records is a typical job and especially if we consider a bunch of data such as managing student data in college. Like student personal details, academic details and skills, resume and even the placement details of each and every student.
* HireUp is an application that wills helps college administrator in managing all these data efficiently and securely with error free and also helps avoids any duplicate entry so that the available space can be properly utilize.

**HireUp – Introduction, Scope, Objective:**

* HireUp is a web-based application developed using Core Java, JSP, and we use MySQL for database management.
* HireUp is a platform where college administrator can manage student data related to placement work. The platform can be access by user (student) and admin (college administrator).
* The user (student) can register to access the application and enter his/her details. Such as personal details, academic details, technical and non-technical skills which are then store in the database.
* System generates a resume using all the entered student data, which can be view by both student and admin.
* Admin further can maintain student placement details such as company name, package detail, joining data.

HireUp Hardware & software Requirement

* Ashish

HireUp functional and non-functional requirement

**Requirement Gathering**

* So here these are the Primary Requirements, which we both have discussed and come up with.

**Explain the modules and its process of how you developed the solution of module. Explain the obstacles you faced in the process and how you solved them.**

Make JDBC connection by creating connection object using the function define in class DbConnection of hireupdb package

NOTE: At each and every page session attribute is collected, this also helps in validating user or errorpage is display.

Pages Logic:

**Index.jsp Header and footer logic:**

* Using header and footer on multiple page using JSP directive tag instead of stand jsp:include since using jsp directive tag the code are added at compilation time while in jsp:include it is done at run time which will put load on performance.
* In header, we have condition to check if session with attribute username or admin, is null or not, if not then for student user, name and logout is shown and same for admin

**Signup Process (**problem: avoid duplicate account)

1. Signup form
   * Enter registration detail (email, username, and password)
2. Signup Action
   * Collect the detail enter in signup form
   * First, check if user already exit or not in login table. By passing query to count(\*) where username = ? **or** email = ?.
   * If count(\*) return 0 (zero), go on to insert data In the login table and register user successfully.

**Forget Password and Reset Password: (**problem: check for correct user, and password should not match with old password)

1. ForgetPassword
   * Ask for username and email registered with that username and submit from redirect to Action page
2. ForgetPasswordActionPage
   * Collect username and email entered
   * Check if username, exist in database using count in select query
   * if yes, validate whether email is registered with given username or not
   * if both condition is yes, redirect for ResetPassword
3. ResetPassword
   * Here, ask user to enter his new password and call ResetPasswordAction page
4. ResetPasswordAction page
   * Now make JDBC connection
   * Collect NewPassword
   * Get OldPassword from database and check if old and new password matches if yes display message accordingly
   * If no, then update the database and display message “Password changed successfully”

**Login process (**problem: allow login only if user if not logged in, and

1. LoginPage

* First check if any session is active or not, by checking whether session attribute is null or not for admin and username. Since if user if already login, there’s no need for login again.
* Else, let user fill the login form and make call to login action page.

1. LoginAction

* Collection username and password enter in login form and make connection to JDBC
* How first check account type if it is admin login or user login, by counting if records with given username exist in admin table or not.
* **If yes,** then its admin user 🡪 validate password, by fetching password from database and comparing entered password. If password matches then go to **admin index** or display error message.
* If no, then its student user 🡪 validate password, for user by fetching password from database and comparing password and if password matches then go to **user/UserLoginAction** or display error message

1. User/UserLoginAction
   * Check if user is existing or new by checking whether details already exist or not or Having incomplete profile
   * Check if user is new, by checking whether the email, registered with given username is store in student personal and academic detail
   * If email is present in both table, then redirect to Dashboard
   * If email is not present in any one of them, then **set email session attribute** and redirect to fill that table attributes value

**Add User Details:**

1. AddUserPersonalDetail
   1. This File represent the backend process of AddUserPersonalDetail.jsp page
   2. First session attribute email is collected, which also validate legal access to this page
   3. Now using email, check if there’s any records already exist with this email, if yes go back to UserLoginAction.
   4. Else, the value parameter is collected from AddUserPersonalDetail.jsp page and this values are enter with current user email id into the hireupv1.student\_personal\_detail database table
   5. Then redirect to AddAcademicDetail page
2. Repeat same process of AddUserPersonalDetail for AddUserAcademicDetail and AddUserSkillsDetail

**View user personal and Academic Details:**

* Session is collected, which also help validate legal access
* Check if user records exist or not, if not then redirect to UserLoginAction else display the details.

**View Resume by user:**

* Session and validate legal access
* Check if user data exist in all of table’s personal, academic, skills. If not then redirect to UserloginAction to check if this user is existing or new or having incomplete profile.

**User Setting page:**

* At here user can update/edit the account detail and also can change password or delete the account all-together if needed. If update is successful or unsuccessful message is shown accordingly.
* 5 different forms are displays, respectively for  
  1. Update personal details  
  2. Update academic details  
  3. Update skills details  
  4. Update password  
  5. Delete account.

**Update personal, academic, skills details, update password:**

* Collect the values from respective form, make JDBC connection, update database and display message accordingly.

**Delete user account:**

* Collect password value from form, and check if the password is matching with user login table data.
* Then delete user records from each tables one by one,   
  1. Delete personal details  
  2. Delete academic details  
  3. Delete skills details  
  4. Delete from company  
  5 finally, delete from login data and redirect to index.jsp page

**UserError page**

* Check if username session is null or not,  
  if null, then 401: unauthorized error  
  else, 400: Bad request error, also display error\_outline using predefined object in jsp web-container i.e exception.getMessage()

**Admin Index and pages**