Java Bootcamp

1. Tax deduction at source (TDS) calculator

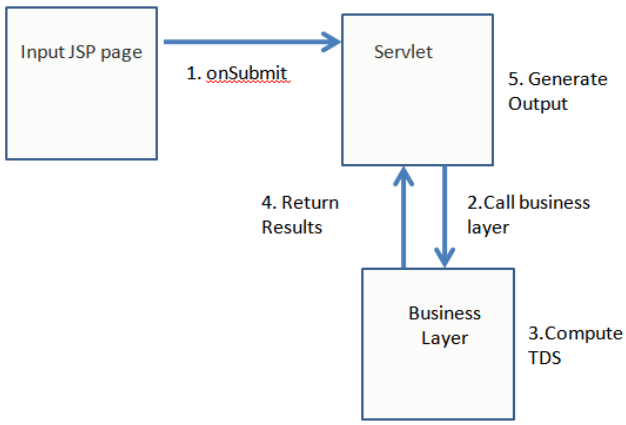
TDS is one of the modes of collection of taxes, by which a certain percentage of amounts are deducted by a person at the time of making/crediting certain specific nature of payment to the other person and deducted amount is remitted to the Government account. It is similar to "pay as you earn" scheme.

Problem Statement:

Write a dynamic web application to calculate TDS for a given salary input.

Input: Monthly salary entered from a single text form field.

Page Flow:



Business Rule:

TDS slabs are:

1) Up to Rs.150000/year -> No TDS (Nil)
2) From Rs.150000 to Rs.300000/year -> 10.3%(TDS rate)
3) From Rs.300001 to Rs.500000/year -> 20.3% (")
4) Above Rs.500000 -> 30.3% (")

For Example: If a person's earning is Rs.28000 per month, TDS should be deducted from his salary.

Output Expected:

TDS Calculation:
Salary per Year: RS.28000*12 = 336000
Up to Rs.150000 no TDS will be deducted.
So, 336000-150000 = 186000 it comes under 1st slab.
Hence TDS for year is 186000*10.3% = 19158
TDS deduction per Month is, 19158/12 = Rs.1596.5
The Net Salary of Mr. X per month is: 28000-1597 = Rs.26403/-

2. Quiz Application

A quiz is a way to test the knowledge of a person.

Problem Statement:

Write a multi user web application to guess the correct answer for a given question. A user is prompted a question and expected answer is displayed using "*".

User of this application can submit only one character at time, each time the user predicts the correct character "*" will be replaced with the guessed character.

Example: if expected answer is "Java" on the screen "****" is displayed. When user submits character "a" answer changes to "*a**", next if he submits "a" once again it changes to "*a*a", until he predicts all the characters in the answer correctly.

Each user has a MAX life line of 5 choices, if the user makes 5 mistakes the user will be redirected to home page with the message "You Lost!!!".

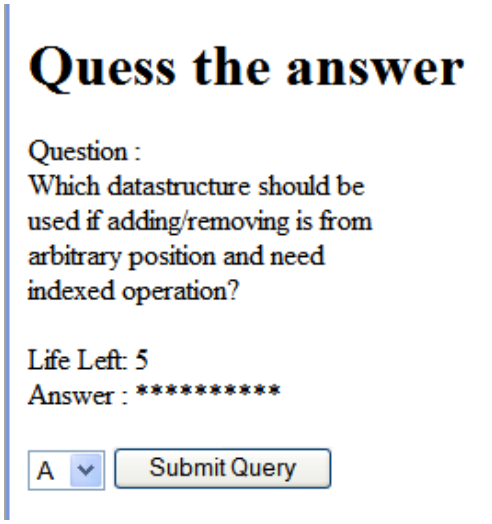
If user predicts the answer correctly user gets the message "You Win"

In both the cases the user can choose to answer next question.

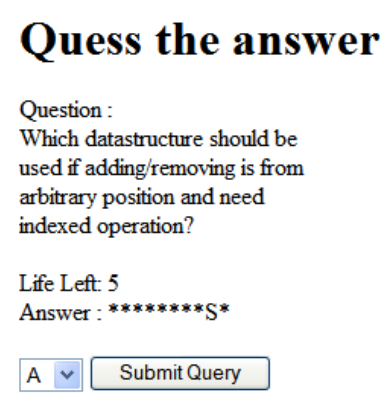
Below are some Screen Shots:

Index page

Clicking on Hyperlink, it redirects to page with question and the user starts with 5 life lines.

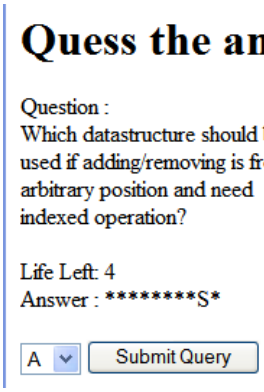


User predicts correct character for answer.



User predicts wrong character

Reduce Life left by one.



After predicting all characters correctly

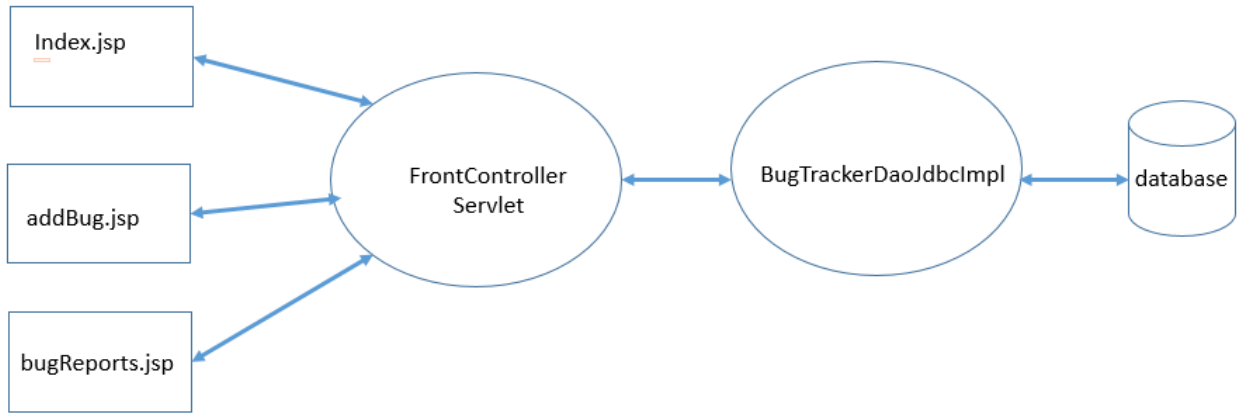
User is redirected to index page:

If user 5 mistakes per question:

3. Bug Tracking System

Design and implement a bug-tracking system for software projects. Project information should be available in the database. A bug has to be reported on this system. While reporting a bug the project against which the bug is raised should be available. From then on, bug reports for the project can be viewed.

Application architecture:



Note:

- The database table required to maintain the employee information has to be created & populated manually through the back-end. Necessary fields should be added to this table, to support all the use-cases listed below.
- Do not implement authentication/authorization.

Create the database tables with proper association between them. The fields which need to be captured are project title, start date of project, end date of project, bug title, bug description, bug severity [“critical” | “Major” | “Minor”] , bug status [“open” | “fixed” | “closed”], bug created on date.

Insert sample data for the created tables.

Use case #1: Report a new Bug	
Pre-Condition	The project against which the bug is raised, should be available.
Trigger	User clicks on “Report a new Bug” from the main screen
Post-conditions	The details of the new bug are added to the application database.
UI Screen Details	The input form comprises of the following fields • Title: text value • Project: Single-select list box, pre-populated with all project names present in the database • Description: multi-line text value • Severity: single-select list, containing the values ‘critical’, ‘major’, ‘minor’.

	• Status: single-select list, containing the values 'open', 'fixed', 'closed' • Submit button • Cancel button
Main Flow	The user provides the required values and clicks on the "Save" button. On successfully saving the record, the user is redirected to the main screen, and the following message is displayed there: "The new bug report has been added successfully".
Alternate Flow	If the user chooses to abort the "add" operation by clicking on the "cancel" button, he should be re-directed to the main screen
Exception flows	If the user provides incorrect or incomplete values in any of the input fields and tries to save the record, the operation is aborted and the following message is displayed on the screen: "The value provided for the field <title|severity|...> is incorrect or incomplete".

Use case #2 : Bug Reports						
Pre-Condition	None.					
Trigger	User clicks on "Bug Reports" from the main screen					
Post-conditions	The system displays a tabular report (shown below) containing all the details of bugs					
	Project Title	Bug Title	Bug Description	Bug Created Date	Severity	Status

Import eclipse "bugTrackingSystem" dynamic web module. The web module contains the following implementations:

a) "Index.jsp" page with hyperlinks for both the use cases. Update the "href" to invoke the Controller [MVC].
b) Completed "addBug.jsp" page.
c) Partially implemented "FronController.java" Servlet Controller class for MVC application. Configure this controller with appropriate URL pattern in deployment descriptor "web.xml".
d) "Project.java" and "Bug.java" entity/model classes
e) "BugReportDTO.java" data transfer object with fields required for Use case #2
f) "BugTrackerDao.java" interface.

Things to Do:

a) Create database tables with proper association between them as per the requirements mentioned. Insert few records for these tables. Provide Database scripts for the same.
b) Update web.xml to configure "FrontController" servlet and provide appropriate URL pattern.
c) Client side validation: Validate user input using JavaScript.
d) Convert Scriptlets code to use JSTL in "addBug.jsp" page.

e) Create JSP page to display “Bug Reports” using JSTL.
f) Create “BugTrackerDaoJdbcImpl.java” implementation class for “BugTrackerDao.java”. This class should contain code to interact with MySQL database as specified in the requirements.

4. Books Information

Develop a web-based application to update information about the books.

Part 4.1

The homepage should display book’s data as shown below; Book’s data is fetched from the already pre-populated in server’s memory.

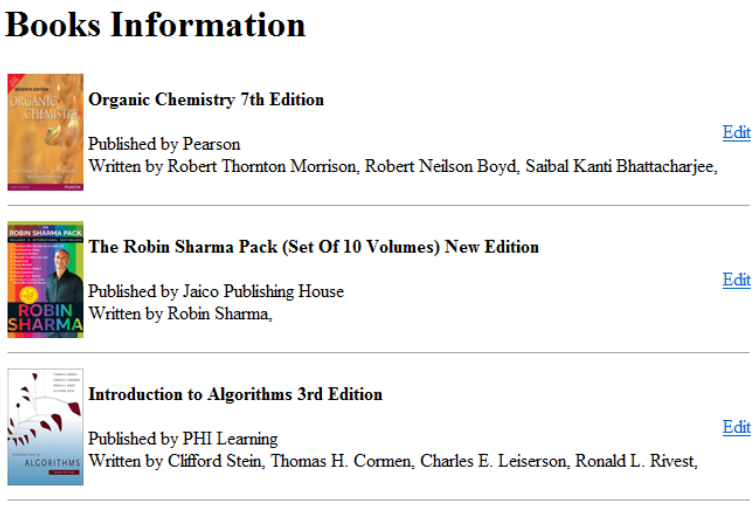


Figure 1: Book Home page

For each book, display the following information:

4.1.1 Display picture of the book.

4.1.2 Display authors using comma separated values as
Written by author1, author2, etc..
For example, for the book “**Organic Chemistry 7th Edition”** the author information should be displayed as “Written by Robert Thornton Morrison, Robert Neilson Boyd, Saibal Kanti Bhattacharjee.”

4.1.3 Display other information as show in the screen shot.

Part 4.2

Edit Book information

When the user clicks the "**Edit"** hyperlink as shown in Part 2.1, the corresponding information should be displayed in an editable form as listed below.

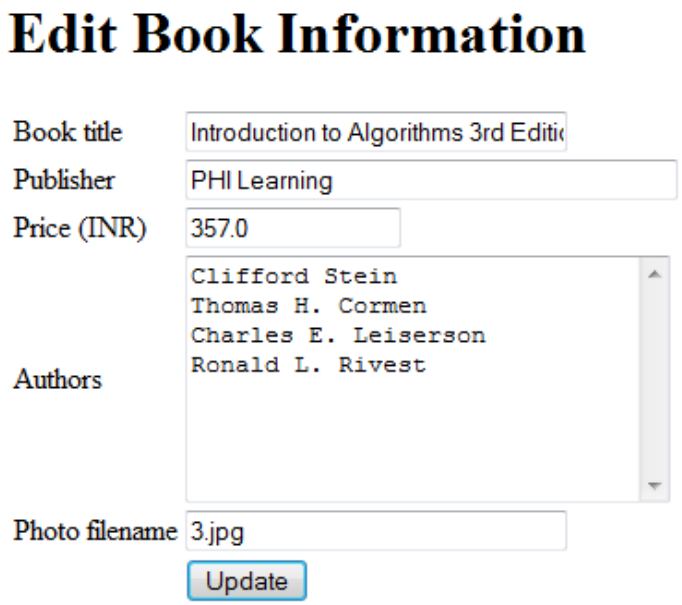


Figure 2: Edit book information

4.2.1 All the fields are supposed to be populated with relevant data.
4.2.2 Authors should be displayed in a text area (one author per line)

Part 4.3

Update information

4.3.1 Once the book's information is modified and on submitting the form information by clicking the "Update" button, the book information should be updated and the user should be redirected to the homepage, where the changes are reflected.

5. Faculty Feedback

Build a faculty evaluation web application for an engineering college. Application should allow students to give feedback for a faculty. The application should generate reports based on feedback provided.

For this application we do not need authentication/ authorization.

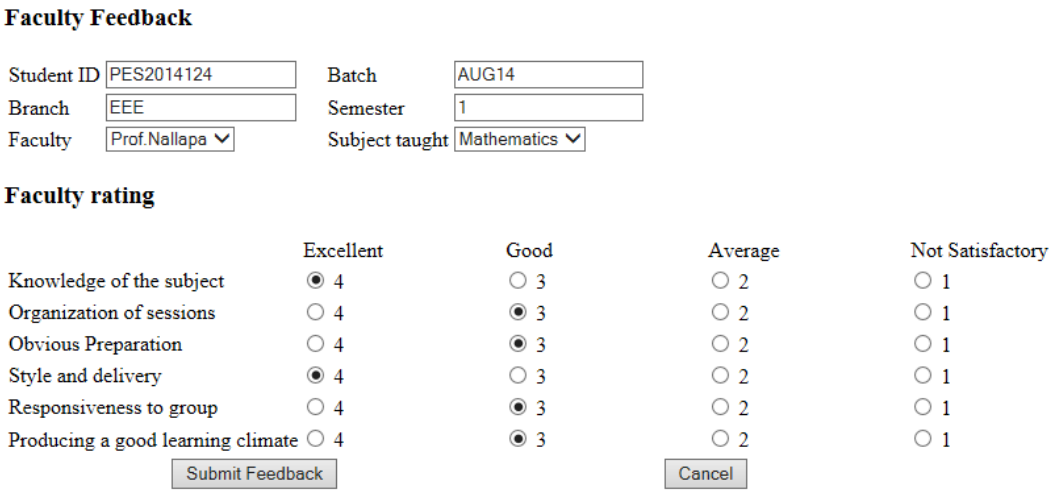
Create the "STUDENT", "FACULTY", and "SUBJECT" database tables. Insert sample records for the same. **Provide DDL and DML scripts.**

What to implement:
Home page should have two links:

a) Provide Feedback
b) View Faculty Report

Screen 1:

Clicking on "Provide Feedback" user is redirected to this page which allows the application user to provide feedback for a faculty.



Components included on this page are described as follows:

a) Student ID: text input of "Student ID" present in "STUDENT" table of database. Once "Student ID" is entered "Batch", "Branch" and "Semester" details should be fetched from database and populated into respective text boxes. **Implement this using AJAX (jQuery) and JSON**
b) Batch, Branch and Semester: text boxes which are populated from data fetched from database based on "Student ID" as explained above.
c) Faculty: dynamic dropdown pre-populated from faculty details present in the database
d) Subject: dynamic dropdown pre-populated from subject details present in the database
e) Radio Buttons: radio buttons for rating on different criteria as shown in the screen

f) "Submit Feedback" button: Clicking on submit button the information has to be stored in the database table and user should be redirected to home page with the message "Details added successfully".
g) "Cancel" button: clicking on cancel button the operation has to be aborted and user should be redirected to home page

Screen 2:

User will be redirected to this page on clicking "View Faculty Report" hyperlink.

Faculty Prof.Nallapa Batch AUG14 Semester 1 Get details

Faculty rating	Excellent	Good	Average	Not Satisfactory	Avg.
Knowledge of the subject	22	5	1	2	3.56
Organization of sessions	15	11	3	1	3.33
Obvious Preparation	24	4	2	0	3.73
Style and delivery	25	5	0	0	3.83
Responsiveness to group	22	7	0	1	3.66
Producing a good learning climate	18	9	1	2	3.43
Average Rating	3.59				

Screen Details:

a) Faculty: dynamic dropdown pre-populated from faculty details present in the database
b) Batch: text box input for which faculty feedback report is required
c) Semester: text box input for which faculty feedback report is required.
d) "Get Details" submit button: On clicking the button the report is displayed as shown.
   - Report should display the number of students who have rated for each criteria. For example in the above report "22" students have rated "Excellent" in "Knowledge of the Subject", "5" have rated as "Good" and so on.
   - Average Calculation details:
     Points are as listed below
     Excellent = 4; Good = 3; Average = 2 and "Not Satisfactory" = 1
     Example for Avg. calculation for "Knowledge of the subjects:
     - Total number of students who have provided feedback = 30
     - 22 students have rated as "Excellent".
     - 5 students have rated as "Good"
     - 1 student has rated as "Average"
     - 2 students have rated as "Not Satisfactory"

     (22 * 4 + 5 * 3 + 1 *2 + 2 * 1) / 30 = 3.56

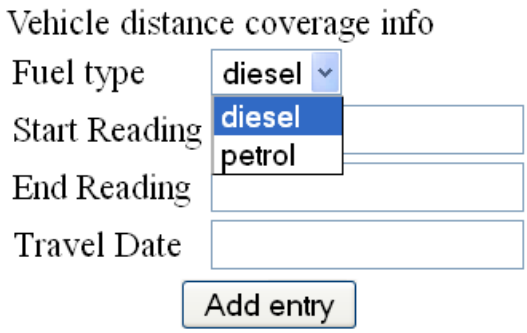
6. Vehicle Fuel expenses

Develop a web based application to compute the total cost incurred on vehicle fuel expenditure.

Part 1.1

A vehicle can run on “Petrol”, “Diesel”, “CNG”, etc. Cost of fuel can fluctuate and hence the cost should not be hardcoded.

Display the home page as shown below:



The drop-down “Fuel type” should be populated from the data stored in servlet context.

On submitting the form as shown above by clicking on “Add entry” submit button, the data has to be stored in appropriate scope and the details added should be listed as shown below.

Note:

1) The data added by each user should be maintained separately.

2) Do not use database for this application.

Sample screen shots for entries by two users:

User 1

Vehicle distance coverage info

Fuel type: diesel

Start Reading: 1290

End Reading: 1320

Travel Date: 11-01-2011

Add entry

Date of travel	Vehicle fuel	Start Reading	End reading	Distance Covered	Total Cost incurred	Delete Entry
10-Jan-2011	diesel	1200	1290	90	4609.8	
11-Jan-2011	diesel	1290	1320	30	1536.6	

User 2

Vehicle distance coverage info

Fuel type: petrol

Start Reading: 8945

End Reading: 8966

Travel Date: 6-02-2012

Add entry

Date of travel	Vehicle fuel	Start Reading	End reading	Distance Covered	Total Cost incurred	Delete Entry
05-Feb-2012	petrol	8900	8945	45	3375.0	
06-Feb-2012	petrol	8945	8966	21	1575.0	

Remove an added entry

Date of travel	Vehicle fuel	Start Reading	End reading	Distance Covered	Total Cost incurred	Delete Entry
05-Feb-2012	petrol	8900	8945	45	3375.0	
06-Feb-2012	petrol	8945	8966	21	1575.0	

When a user clicks on Delete Entry, the entry has to be deleted and the remaining entries of that user have to be displayed.

7. Task Management

Task lists and tasks help you easily keep track of every job on a project, quickly telling you who, when and why.

Build a web application using which you can assign tasks to staff, co-workers and contractors in seconds. Application should allow the process of adding and assigning tasks and sub-tasks, and allows you to assign tasks to several people instead of just one.

For this application we do not need authentication/ authorization.

Create a table to store **employee** and **project** information in the backend.
Employee can work in a single project, and project can have many employees.
Insert required records. Table details are listed below. **Provide DDL and DML scripts for the same.**

What to implement:
Home page should have two links:

c) Assign tasks
d) Task view

Screen 1:

Clicking on "Assign tasks" user is redirected to this page which allows the application user to add tasks for employee who are working on a given project.

Components included on this page are described as follows:

h) Project: dynamic drop-down populated from project data present in database. On selecting "Project" all employees working in this project should be populated in list "Who should do this?" **Note: Implement this feature using AJAX (jQuery)/JSON.**
i) Description: description text of the task
j) Start date of task: Date when the task is assigned to employees. [DD-MM-YYYY] format
k) Due date of task: date when the task has to be completed. [DD-MM-YYYY] format
l) Who should do this? : Multi-select list of employees to whom the task is assigned, Note: As explained above this list is dynamically populated when a project is selected.
m) "Submit" button "Add a task". Clicking on submit button the information has to be stored in the database table and user should be redirected to home page with the message "Details added successfully".
n) "Cancel" button: clicking on cancel button the operation has to be aborted and user should be redirected to home page

Tech College

Assign Tasks

Project *	iPhone app UI
Description *	Push Notifications
Start Date of Task [dd-mm-yyyy] *	10-1-2015
Due Date of Task [dd-mm-yyyy] *	12-2-2015
Who should do this? *	Rahul Dev M1001075 Apoorva R M1001089 Anjan G M1001099 Bhushan S M1001119

Add a task | Cancel

Screen 2:

User will be redirected to this page on clicking "Task View" hyperlink.

Screen Details:

e) Filter by Project: Dynamic drop-down of all projects present in "PROJECTS" table.
On selecting project the task details assigned for that project should be displayed as listed above.
If "All Projects" is selected, tasks assigned to all the projects should be displayed.

f) Back hyperlink: user is redirected to Home page

View Tasks

Filter by Project All projects

Project: iPhone UI

Task Description: Push Notifications
Task Start Date: 10-1-2015
Task End Date: 12-2-2015

MID	Employee Name
M1001075	Rahul Dev
M1001099	Anjan G
M1001119	Bhushan S

Task Description: Help Screen
Task Start Date: 15-1-2015
Task End Date: 22-2-2015

MID	Employee Name
M1002000	Kavitha B
M1001072	Rohit Reddy

Project: iPad Bugs

Task Description: Graphics Slicing
Task Start Date: 5-1-2015
Task End Date: 10-1-2015

****

8. Employee Satisfaction Survey

Create a happier workplace using employee satisfaction survey. Happy employees are more likely to delight co-workers, customers, and partners and build the good reputation of your brand. Use employee surveys to check in with employees about their satisfaction with their roles and responsibilities, the work environment, and their experiences with management, etc.

The survey will have questions as listed below, each question can have minimum of two options, and maximum number of options is not limited:

1. How meaningful is your work?

- ○ Extremely meaningful
- ○ Very meaningful
- ○ Moderately meaningful
- ○ Slightly meaningful
- ○ Not at all meaningful

2. How challenging is your job?

- ○ Extremely challenging
- ○ Very challenging
- ○ Moderately challenging
- ○ Slightly challenging
- ○ Not at all challenging

Pre-condition for the application is that tables should be created for

- Employee with columns (employee ID, name, and email) in the backed and data has to be inserted.
- Tables for storing survey question and options

Provide all the DDL and DML database scripts used for creating tables and records inserted. Insert at least 3 employees and at least 4 survey question with options.

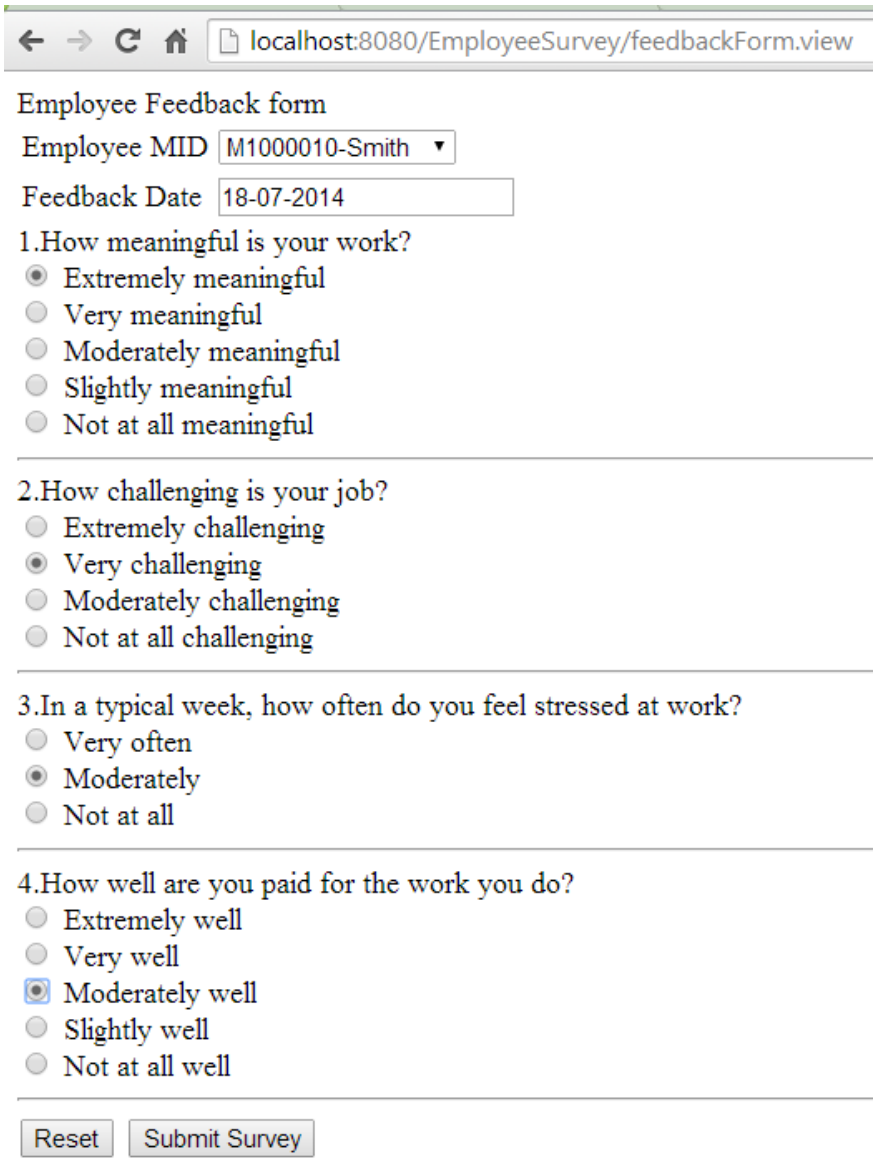
Note: The tables should be in 3NF.

What to implement:

Screen 1: Display Survey Form

This screen will allow employees to provide feedback as shown below:

Tech College



Screen details:

a) Single select pre-populated with all employees present in the database should be displayed with MID-NAME.
   - On selecting the Employee "MID-NAME" from dropdown if the employee has already provided the feedback the screen should be pre-populated with existing data present in the database. Implement this feature using AJAX with JQuery library.
   - Employee can change the feedback details

b) Feedback Date should be pre populated with system date.

c) All survey Questions and its Options should be displayed.

d) "Submit Survey" button. Clicking on it the information has to be stored in database and user is redirected to home page with message "Survey details are submitted successfully" should be displayed on home page.
   - If it's a new entry record should be added into the database table
   - If employee modifies the survey and clicks "Submit Survey" button, details should be updated in the database.

e) "Reset" button clears the selected options

f) Provide “Back” hyperlink [Not shown in the screen] to redirect user back to home page.

Screen 2:

Employee Survey Report

The report shows the survey projections for the current calendar year [example: 2014]. Report should show all survey questions and options along with number of employees who have selected each option

Below is a sample screen for two survey questions.

Employee Survey Report

Year	**2014**
Feedbacks Received	**100**

#	Survey Question	Options	No of employees who selected this option
1	How meaningful is your work?		
		Extremely meaningful	28
		Very meaningful	12
		Moderately meaningful	40
		Slightly meaningful	15
		Not at all meaningful	5
2	How challenging is your job?		
		Extremely challenging	19
		Very challenging	31
		Moderately challenging	36
		Not at all challenging	14

9. Bid application

This application offers people to sell their land (site) via auctions.

Assumptions for the web application: Users and “site details” should be already be available in the database. User should have “user name”, “email”, “address” and “password”.

A “Site” will have a seller and Buyer along with site details like “address”, “dimension of site”, “minimum bid value” and “reserve value”. User of application can be “Seller” and/or “Buyer”.

Note: Create database tables for 3NF and provide DDL and DML for the same.

The home page should contain links to

a) Login.
b) Bid options

What to implement:

Screen 1: Login.

User Name : text

Password: text

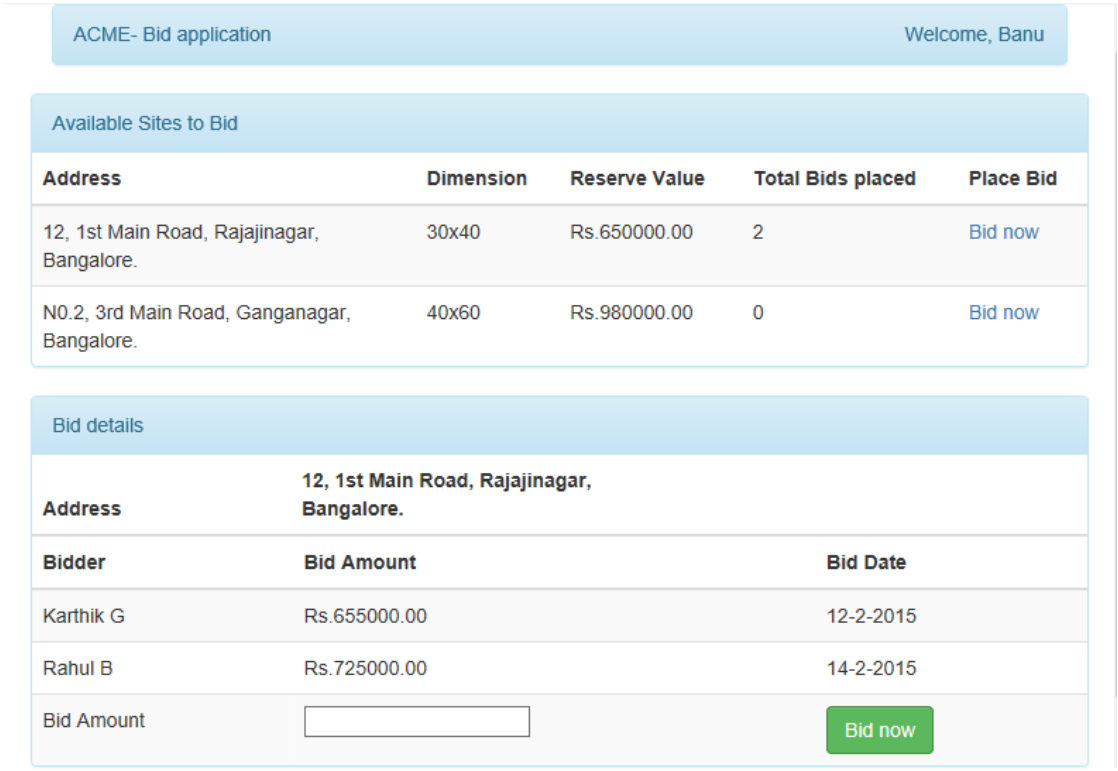
Login Button

User should login using “email” and “password”. If User is a valid user he/she will be redirected to bid screen as shown in screen-2.
If User enters invalid credentials “invalid user/password” should be displayed on the “login screen”.

Screen 2: Bid screen

Screen should display

a) Welcome, <<user name of person who has logged>>.
b) All bids which are open. Details should contain “address”, “dimension”, “reserve value”, “total bids placed for that site” and a hyperlink clicking on which a form should display showing previous bids and to accept bid amount as shown below:



- Once bid amount is entered the bid details should be stored in the database. The bid details should contain the person who has bid and the amount. Once Bid is placed "total bids placed" count should be updated and bidder information also has to be displayed along with previous bidders
- Implement all functionalities of screen 2 using AJAX (jQuery) and JSON.