**Read the below instructions carefully before you start.**

* Software Requirement: NetBeans 8.\*, MS SQL Server 2008 R2+
* Download the given material then unzip.
* You receive folders: Q1\_XXX (1) and Q2\_XXX (2) … They are template for your corresponding exam questions. Q1\_XXX used for question number 1 and Q2\_XXX for question number 2 and so on. Open them as NetBeans project, replace XXX by [your student ID], e.g. Q1\_SE1111, Q2\_SE1111, Q3\_SE1111. Change the context application to new name, e.g. /Q1\_SE1111/Index.html instead of /Q1\_XXX/Index.html. **Your work will not be considered for grading, if this condition is not satisfied**.
* Template project already contains the **DBContext.java** (3) to source package (**src\java\dal**) under folder (1),(2),.... You might need to rename a java package for (3), free to set the name of package.
* You **MUST** use **DBContext.java** for your all database connection. Update your database connection string by filling values for instance variables in the constructor.
* All extra libraries (if needed) must be used by available NetBeans libraries only.

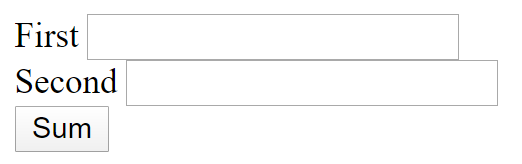
On completion, read submission guide carefully.

**Question 1 (2 points)**

You are given a NetBeans template project (read more detail in the first page). The given project already added index.html.

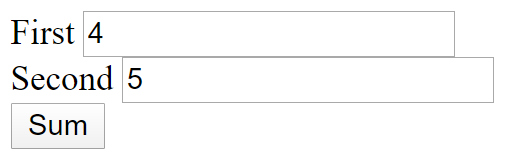
Your tasks

* Edit index.html to create a web page (**0.5 point**)



* Assume that users enter some digits to the text fields. He/she clicks [Sum] button then the browser makes a request to /compute using Post method **(0.5 point)**.
* /compute is URL to access Servlet that displays the SUM of First and Second **(1 point)**

see example:



* Servlet responses **9**

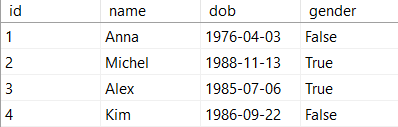
**Question 2 (3 Points)**

You are given a database script contains [Student] table. Your tasks:

* User can access **/search** to search for students. The system displays as following: **(0.5 point)**



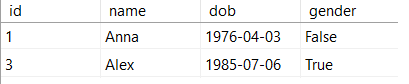
* User can click [Search] button
  + If [Name] contains no character 🡪 system displays all students **(1 point)**



* + If [Name] contains some characters 🡪 system displays the students who have name start with [Name] **(1.5 point)**. Example:

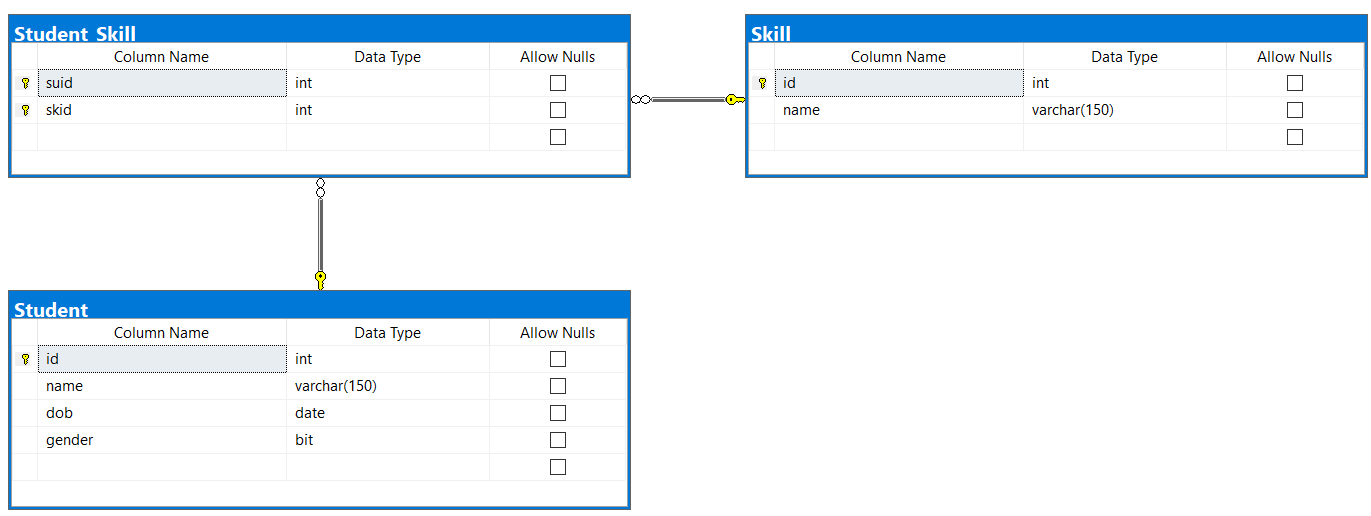


System displays:



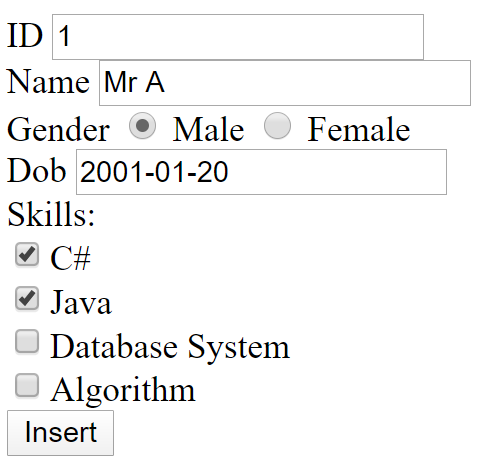
**Question 3 (5 points)**

You are given a database



Hint: Student and Skill is many-to-many relationship 🡪 Student\_Skill created to represent it.

User can access **/insert** to add new Student with his/her skills as following:



* The list of skills loaded from [Skill] table **(1 point)**
* When user clicks [Insert] button
  + 01 student added to [Student] table **(1 point)**.
  + Corresponding skills registered to [Student\_Skill] table **(3 points)**.