**PRJ321’s POLICIES**

1. **ASSIGNMENT EVALUATION**
   1. **Coding Convention**

* Variable name, the function name must be meaningful.
* Package names must be lowercase
* The class name must start with the noun, the first letter in uppercase and then use the camel rule to write the remaining components. For example: SinhVien.
* The function name must start with the verb, the first letter in lowercase, then use the camel rule to write the remaining elements. Example: calculateMark ()
* The variable name must be a noun that begins with a lowercase letter, then use the camel rule to write the remaining elements, the next words starting with a capital letter. Example: phoneNumber
* The constant name must be all capital letters, separated words by the underscore character "\_". Example: MAX\_INTERATION
* Ensuring the basic principles of OOP, inheritance, methods of accessing object properties
  1. **Cover Requirements**
* Meet all the requirements.
* Regarding the interface, meet the requirements of the alignment, the display format, the layout.
* All data on the web is loaded from the database, or according to specific requirements in the assignment.
  1. **Cover Common Requirements**
* Do not import redundant libraries. Do not import like java.util. \*
* Do not declare redundant variables, functions.
* After opening the connection to the DB, the connection must be closed.
* Accessing data only takes data that is needed, not redundant.
* All SQL statements must be explicit, specify the column name, do not use SELECT \*.
* ID rules: do not duplicate, cannot be null.
* All data must be validated before storing into the database.
* For database-related assignments, the DB script file must be attached..
* Principles of database design: all tables must have key. Ensure the principles of relational database.
* All errors must be logged in the server's log or create own log file.
* The website name must have a meaningful name.
* The program must ensure that there is no error during running

1. **GRADING PROCESS**

There are 3 rounds:

* 1. **Round 1**: Process checking- **Students who violate the following rules will receive 0 MARK for the whole project**
* Students must understand the data flow, where the data goes, how the data is handled
* Mandatory use of MVC2- design pattern
* Must build the application in the direction of components (daos, dtos, controllers, utilities, ....)
* A diagram is required: block or sequence can be selected
  1. **Round 2**: Code checking
* Each error in section II.1 is deducted 20% of the assignment's MARK.
* Any error in section II.2 will get 0 MARK.
* Each error in section II.3 is deducted 50% of the assignment's MARK.
* Required validate: insert, update: If not validate subtract 50% of the function's MARK.
* There is a fraud in the learning / exam process: copy each other, the default will be 0 MARK for the whole project no matter where you copy from.
  1. **Round 3**: Re-code
* **After all, students have to re-code a module or a part as required by the lecturer or add new features as required by the lecturer. If you can't code again, the whole project will get 0 MARK**.

**Note**:

\* All assignments must be submitted to LMS- on time, late submission is not accepted.

\* You are required to build your product in accordance with the real requirements.