# Statistical Analysis and Data Visualization application

# IDS 517: s18t23

# Submitted by Vidhi, Sumeet and Ishaan

This is an application to perform Descriptive Statistical Analysis and Visualization. The application enables user to upload/select a file, run statistical analysis and visualize patterns through graphs. The functionalities include accepting a valid user's selected file, querying within the file and displaying appropriate results in form of visual graphs.

## Tools, Softwares and Libraries requirements

* Any IDE’s may be used to build the application(Eclipse,Netbeans) .However the framework should be JSF 2.2 and the webpages should be built on jsp. Javascript should not be used.
* MVC, JSF2.2, Java 7 (1.7) or 8 (1.8), Tomcat 8.5 or higher, Apache MyFaces and Tomahawk jsf libraries, Apache Commons Math, and JfreeChart.

## MVC Model

Model–view–controller (**MVC**) is an architectural pattern commonly used for developing user interfaces that divides an application into three interconnected parts. This is done to separate internal representations of information from the ways information is presented to and accepted from the user.

The MVC framework includes the following components:

* **Models**

Model objects are the parts of the application that implement the logic for the application's data domain. Often, model objects retrieve and store model state in a database. For example, a Product object might retrieve information from a database, operate on it, and then write updated information back to a Products table in a SQL Server database.

In small applications, the model is often a conceptual separation instead of a physical one. For example, if the application only reads a dataset and sends it to the view, the application does not have a physical model layer and associated classes. In that case, the dataset takes on the role of a model object.

* **Views**

Views are the components that display the application's user interface (UI). Typically, this UI is created from the model data. An example would be an edit view of a Products table that displays text boxes, drop-down lists, and check boxes based on the current state of a Product object.

* **Controllers**

Controllers are the components that handle user interaction, work with the model, and ultimately select a view to render that displays UI. In an MVC application, the view only displays information; the controller handles and responds to user input and interaction. For example, the controller handles query-string values, and passes these values to the model, which in turn might use these values to query the database.

Flow of Data in the Application (MVC Model)

