In this section we are discussing in details of the proposed work to be carried out for investigations of implementation of web applications with different implementation techniques.

2.1 The Objective

The main objective of the present work is: To make detailed investigations of different implementation techniques for building Web applications. The implementation choices under investigations are: (a) the .NET built by Microsoft Corporation and (b) the J5EE, built by Sun Microsystems and other industry players.

We will make details comparative investigations of the following issues:

(i) The performance of the techniques.
(ii) The efficiency of the techniques.
(iii) The scalability of the techniques.
(iv) The system cost of the techniques.
(v) The stability of the techniques.
(vi) The reliability of the techniques.
(vii) The performance parameters of the applications, such as response time, throughput, hits/sec.
(viii) To verify whether hits/sec and throughput have individual as well as combined affect on response time.

Two prototype research Web applications are developed and tested to study the feasibility of the present work. We have developed one of the Web applications with proprietary Microsoft .NET Platform. The Microsoft .NET is largely a rewrite of Windows Distributed Internet Application (DNA), which was Microsoft’s previous platform for developing enterprise applications. The Windows DNA includes many proven technologies that are in production today, including Microsoft Transaction
Server (MTS) and com+, Microsoft Message Queue (MSMQ), and Microsoft SQL Server database. The new .NET Framework replaces these technologies and includes a WS layer as well as improved language support [26, 27]. We call this Web application as PReWebD.

The other Web application has been developed with open source tool. The benefits of open source tool include the ability to view the source code, to see how features are implemented and to modify the source code to make changes. This allows the developers to port tools to other operating systems and to built new product from open source. A key theme with open source is its flexibility, providing development organizations with source code and the right to modify it. We deploy Java Development Kit (JDK) for the development of the second Web application. The Web application is developed by using Integrated Development Environment (IDE). The IDE is used to develop the Java application for Web application. The NetBeans platform is used to develop the Web application, because this platform allows tool builders to independently develop tools that integrated with other development tools [28]. We call this Web application as PReWebN.

2.2 The Architecture

Two different architectures are used for development of PReWebD and PReWebN. These architectures are discussed in details below.

2.2.1 The architecture of PReWebD

The architecture of PReWebD is shown in fig.2.1. It is 3-tier architecture. This architecture has the advantages of separation, independence and re-usability [29]. The layers of this architecture are presentation layer (PL), business layer (BL) and data layer (DL).

The PL contains the user interface and presentation code. Since the .NET platform is used for design purpose, it includes Active Server Pages Extended (ASPX) pages, user controls, Server controls etc. The BL is the most essential layer in the architecture. This layer is further divided into two sub layers: (a) the Business Logic Layer (BLL) and (b) the Data Access Layer (DAL). The BLL holds the required
Fig. 2.1: The Architecture of the PReWebD
Business Logic (BLo), functions, validations and calculations related to the data. The DAL is responsible for accessing data and forwarding it to BLL. The BL acts as a mediator between the PL and the DL. In ASP.NET sqlClient acts as the data access layer.

The DL manages the physical storage and retrieval of data. It receives the data from the BL and sends it to the database and vice versa. The database queries have been written to access the data from database or to perform operations like insert, update and delete. In our present work, the SQL Server Express 2005 acts as the DL.

### 2.2.2 The architecture for PReWebN

The architecture of the PReWebN is shown in fig.2.2. The JSF provides a component-centric Application Programming Interface (API) from which Web application UI can be assembled. The JSF specification defines a set of basic UI components that can be extended to achieve more specialized behavior. The events from client-side UI controls are dispatched to JavaBeans models which provide Server-side application behavior. In JSF, the UI components are loosely coupled to Server-side Java Plain Old Java Objects (POJO) which is declared as managed beans. The front end Controller servlet handles all Faces requests and dispatches them with the necessary application data, to the appropriate view. The database manages the physical storage and retrieval of data. It receives the data from the model and sends it to the database and vice versa. The database queries have been written to access the data from the database or to perform operations like insert, update and delete. In our present work, the MyStructured Query Language (MySQL) acts as the data layer.

The next chapter of the thesis describes in details about the different tools used for design, implementing and testing of the PReWebD and PReWebN.
Fig. 2.2: The Architecture of the PReWebN

The Model consists of managed beans.

The Faces Servlet acts as the Controller

The View consists of JSP pages.

Fig. 2.2: The Architecture of the PReWebN