The project is conducted **individually**

The objective is to develop **your dynamic, database supported**, web site:

- Choose an application domain: music, trekking, soccer, photography, etc.
- Manage items (music tracks, trekking paths, soccer matches, cameras, ...) and users of the application
- Identify the functionality (extending the base functionality describe later)
- Enable users to access items (search, select, comment) and provide new items

All the techniques illustrated in the lectures must be properly applied (*not a simple, static HTML-based web site*)

The project result is a **running system** and a written **report**.
Structure of the Project

- The **application** must run on the application server that we shall indicate in the labs.
- The **report** must describe clearly in **min 2000, max 3000** words, plus images.
  - The **functions** of the web application and their **motivation**
  - The **architecture** of the application (modules and their roles) – use figures.
  - Main **classes** and main **methods**
  - Major **technical problems** found during the work.
- The project will be **evaluated** according to: complexity of the implemented functions, user interface usability and completeness, organization of the code, coverage of the required technologies.
Functions

- **User Management**
  - List existing users of the system
  - Creation of a new user
  - Deletion of the existing user
  - List and modify access rights of the users - check boxes with some capabilities (min 3) – e.g.
    - A user can comment all items
    - A user can download an item with label “parties”
  - User registration and login to the system
Functions (II)

- **Items management**
  - Users add, edit or remove items
  - Users comments or reviews items
  - Administrator can manage the comments (edit, remove, add)

- **Personalization**
  - Salutation for a returning user
  - List resources that are new from the last visit
  - Customization of the layout for a class of users.
Techniques – MUST be used

- Static HTML: "natürlich"
- CSS: all the look and feel must be in CSS files
- Javascript: check input and manage menus
- Servlet
  - Reading (parameters and headers) and writing headers and resulting page
  - Session management with cookies and session object
  - Redirect the client
  - Forward to another page or servlet
- JSP
  - Expressions, scriptlets and declarations
- Beans
- DBMS access through JDBC
- Integration of JSP and Servlets (forward and include) using MVC pattern.
Software Components

- DBMS – PostgreSQL
  - (on lamj.inf.unibz.it)
- Application server – Tomcat 6.x
  - (http://lamj.inf.unibz.it:8180/)
- Programming language – Java 1.6
- IDE – Eclipse or Netbeans
- Minimal Framework – we will provide you with a minimal framework:
  - Connection to DBMS
  - JSP and Servlet examples
Project Evaluation

- Usability
  - Minimum level of usability! I should be able to visit all the system functions and use them without any "manual" – the navigation through the functions should be supported
- No errors
  - The system should run smoothly without errors and with reasonable response times
- All the required functions should be supported
- All the required techniques should be used
- The report
  - must clearly illustrate the design choices: functions and the technical implementation
  - min 2000, max 3000 words, plus images.
Timeline

- Written exam is on ??? Check the calendar
- The projects should be uploaded not later than ??? Check the course web site
- Please try to upload the project ASAP to see if there are problems
  - Contact lab instructors if you need support
Deployment

- The report should be included together with the project as a PDF file. **In the project main page there must be the link to the pdf**

- **The project must be running on the Tomcat** (lamj) server
  - DB: [lamj.inf.unibz.it](http://lamj.inf.unibz.it) (see next slide)

- Source Code
  - Eclipse users: export war file including sources
  - Netbeans users: zip the project directory (with sources)

- **You must also send the war (or zip) – including the source code! – to:** fricci@unibz.it, and cc: Mehdi.Elahi@stud-inf.unibz.it, cavada@ectrlsolutions.com
Deployment (DB)

- You should use postgresql database on lamj.inf.unibz.it
- Use the same credentials as to access Tomcat server:

```java
public static String className = "org.postgresql.Driver";
public static String url = "jdbc:postgresql://localhost/db1";
public static String username = "user1";
public static String password = "password";
```

- DB name has form db\text{i}, where \text{i} is the same number as used for the username
  - db1 for user1, db2 for user2, etc.
- You can administer your database on http://lamj.inf.unibz.it/phppgadmin/
- You cannot connect to the database from outside the lamj server
E-Mail

- Before the deadline you have to send the email to fricci@unibz.it, cavada@ectrlsolutions.com, Mehdi.Elahi@stud-inf.unibz.it specifying:
  - Your name, surname and student ID
  - Attach the WAR file of the project (or zip for NetBeans)
  - The name of the project
  - The link to the context under lamj:
    - For example, http://lamj.inf.unibz.it:8180/MyProject/
  - All the usernames and passwords we need to test the system