# Teamwork Project Assignment

Design and implement a **Blog** / **Forum** / **Photo Album** / **Audio Album using PHP and HTML / CSS / JavaScript**. Your project must meet all the requirements listed below.

## Requirements

* **Use PHP** – the major part of your work should be PHP written by your team
  + You must additionallyuse **HTML5, CSS3** to create the content and to stylize your web application
  + You may optionally use **JavaScript, jQuery**
  + You may optionally use **PHP frameworks** to simplify your work
* Publish your project **live in Internet** – your project should be public in Internet
  + You may **share your project** to get external feedback
  + Most shared and commented projects will get additional **bonus score**
* **Valid and high-quality PHP, HTML and CSS**
  + Validate (when possible) your HTML (<http://validator.w3.org>) and CSS code (<http://css-validator.org>)
  + Follow the best practices for **high-quality PHP, HTML and CSS**: good formatting, good code structure, consistent naming etc.
  + Follow the PHP best practices: <http://www.phptherightway.com>.
* **Usability and UX**
  + Your web application should be easy-to-use, with intuitive UI, with good usability
* Target all major **modern browsers** and **platforms**
  + Use **PHP5**
  + Ensure your web application works correctly in the latest HTML5-compatible browsers: Chrome, Firefox, IE, Opera, Safari (latest versions, desktop and mobile versions)
  + You do not need to support old browsers like IE9

## Forbidden Techniques and Tools

Using **CMS / blog systems** (like WordPress, Drupal and Joomla), **forum systems** (like phpBB), **photo album system**s (like Plogger) and **audio album systems** (like kPlaylist) is forbidden.

## Project Description

Please choose one of the projects below.

### Blog

**Required** functionalities:

* **View** all posts (optionally with paging).
* Adding **new posts** by the blog owner (after login or password protected). Each post must have **tags**.
* Adding **comments** for every post by visitors – each visitor must fill out his name, email (optionally) and comment text.
* Implement a sidebar holding a **list of posts** sorted by month / year / etc. and a list of the **most popular tags**.
* **Counter of visits** for each post.
* Functionality for **searching** by tags.

**Optional** functionalities:

* Use a **database** (like MySQL or MongoDB) or cloud-based data storage (like MongoLab and RedisLab).
* **User registration** and user profiles.
* **Admin panel**: add / edit / delete posts, comments, tags, etc.

### Forum

**Required** functionalities:

* **View** all questions / categories (optionally with paging).
* Implement a simple **registration** for forum users.
* Adding **new question** by the forum users. Each question must have **tags** and **category**.
* Implement **categories** for the forum questions.
* Adding **answers** to the questions by the forum visitors – each visitor must fill out his name, email (optionally) and comment text.
* **Counter** for visits for each question.

**Optional** functionalities:

* Use a **database** (like MySQL or MongoDB) or cloud-based data storage (like MongoLab and RedisLab).
* **Admin panel**: add /edit /delete forum posts, tags, answers, categories.
* Functionality for **searching** by question, answer and tags.
* Implement **ranking** according to user activity.

### Photo Album

**Required** functionalities:

* **View** all albums / categories (optionally with paging). **Browse** album photos.
* **Creating** new album in a category.
* **Uploading photos** (validating pictures size and type) / **downloading** photos.
* Adding **comments** to photos and albums.
* Implement album's **ranking system** (e.g. vote from 1 to 10 or like / dislike).
* Show the most **highly ranked** albums in a special section at the main page.

**Optional** functionalities:

* Use a **database** (like MySQL or MongoDB) or cloud-based data storage (like MongoLab and RedisLab).
* Implement **user registration**.
* Functionality for **searching** by album name / category.
* **Admin panel** (if registration is implemented): add / edit /delete albums, photos, comments.

### Audio Album

**Required** functionalities:

* **View** all playlists / genres / songs (optionally with paging).
* **Listening** to songs online. **Downloading** songs.
* Creating **new playlist**.
* **Uploading songs** (validating file size and type).
* Adding **comments** to songs and playlists.
* Implement playlists' and songs' **ranking system**. Show the most **highly ranked** playlists in a special section at the main page.

**Optional** functionalities:

* Use a **database** (like MySQL or MongoDB) or cloud-based data storage (like MongoLab and RedisLab).
* Implement **user registration**.
* Functionality for **searching / filtering** by playlist name / song name / genre.
* **Admin panel** (if registration is implemented): add / edit /delete songs, playlists, genres, comments.

## Deliverables

Put the following in a **ZIP archive** and submit it (each team member submits the same file):

* The complete **source code** of your project (PHP, HTML, CSS, images, scripts and other files).
* A **presentation** of your project (e.g. PowerPoint slides) of your project. It should provide the following information:
  + Project name and purpose – what you have created?
  + Team name, list of team members.
  + Contributions of each team member.
  + Technical description.
* Any other information (optionally).

## Assessment Criteria

* **Functionality** (all the required functionalities according to the type of project you choose) – **0…12**
* **UI** (HTML / CSS / Usability / UX) – **0…2**
* **Code quality** (correct naming, code formatting, separation of concerns, etc.) – **0…3**
* **Bonus** (bonus point are given for implementing optional functionalities according to the type of project you choose) – **0..5**

## Live Project Defense

Prepare to present your project live, to demonstrate it and explain how it works.