project: Blog

by Wojtek & Axel

Collaboration :

Result:

Axel reviewing Wojtek :

- DOCUMENTATION: 3/4,
- ORGANIZATION: 3/4,
- HTML: 4/6,
- CSS: 4/4,
- PHP: 4/12,
- MYSQL: 3/8,
- MVC: 2/8
- GIT: 3/4,

Personal comments: (can be removed)
Different knowledge level vs time remaining to finish the project makes learning difficult and has nothing to do with co workers willing to collaborate.

Wojtek reviewing Axel:

- DOCUMENTATION: 4/4,
- ORGANIZATION: 3/4,
- HTML: 5/6,
- CSS: 3/4,
- PHP: 8/12,
- MYSQL: 7/8,
- MVC: 6/8
- GIT: 4/4,

Personal comments: (can be removed)
Good communication.
Good support.
Good analyzing each part of project.
Bad planning organization

Project Information

In this project we are going to develop step by step a blog.

To achieve this we will combine HTML5, PHP and MySql.

The posts of this blog will have an author, so a login & register page. But overall will have basic functionality. Meaning that the main page will only have the functionality of displaying and creating posts.

For the database, this is going to just track the comments, users and it's posts.

Objectives:

General requirements.

- You will not use any libraries for PHP backend.
- Deadline is Tuesday 14th January (one week).

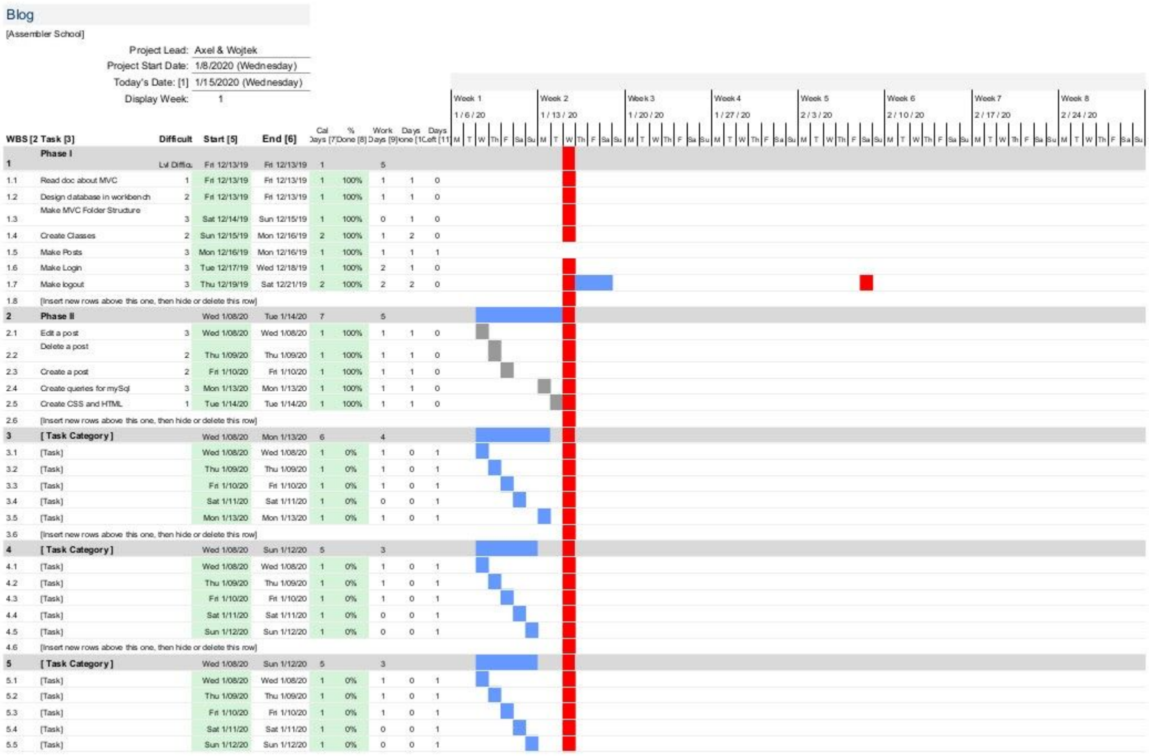
Requirements are for the main page.

- A user will have the option to login. So logout option must be also implemented.
- A user can comment a post anonymously.
- A user can get a link to be able to share a post.
- A user can search for a post by title and contents.
- A user can create posts (only when logged in).
- A user can define keywords for a post.
- A user can publish a post later, in the meanwhile that post will be draft.
- A will can define the date of the post. (If this date is future, the post will not show up until that date).
- A user will define a category for a post.

Requirements for the control panel page:

- A user can login/logout. (will not show panel functionality until log in).
- A user can see it's posts.
- A user can edit a post.
- A user can delete a post.
- A user can create a category.
- A user can edit a category.
- A user can delete a category.

Task list:



Blog

[Assembler School]

Project Lead: Axel & Wojtek
Project Start Date: 1/8/2020 (Wednesday)
Today's Date: [1] 1/15/2020 (Wednesday)
Display Week: 1

WBS [2	Task [3]	Difficult	Start [5]	End [6]	Cal Days [7	% Done [8]	Work Days [9]	Days Done [1C	Days Left [11]
1	**Phase I**	Lvl Difficu	Fri 12/13/19	Fri 12/13/19	1		5		
1.1	Read doc about MVC	1	Fri 12/13/19	Fri 12/13/19	1	100%	1	1	0
1.2	Design database in workbench	2	Fri 12/13/19	Fri 12/13/19	1	100%	1	1	0
1.3	Make MVC Folder Structure	3	Sat 12/14/19	Sun 12/15/19	1	100%	0	1	0
1.4	Create Classes	2	Sun 12/15/19	Mon 12/16/19	2	100%	1	2	0
1.5	Make Posts	3	Mon 12/16/19	Mon 12/16/19	1	100%	1	1	1
1.6	Make Login	3	Tue 12/17/19	Wed 12/18/19	1	100%	2	1	0
1.7	Make logout	3	Thu 12/19/19	Sat 12/21/19	2	100%	2	2	0
1.8	[Insert new rows above this one, then hide or delete this row]								
2	**Phase II**		Wed 1/08/20	Tue 1/14/20	7		5		
2.1	Edit a post	3	Wed 1/08/20	Wed 1/08/20	1	100%	1	1	0
2.2	Delete a post	2	Thu 1/09/20	Thu 1/09/20	1	100%	1	1	0
2.3	Create a post	2	Fri 1/10/20	Fri 1/10/20	1	100%	1	1	0
2.4	Create queries for mySql	3	Mon 1/13/20	Mon 1/13/20	1	100%	1	1	0
2.5	Create CSS and HTML	1	Tue 1/14/20	Tue 1/14/20	1	100%	1	1	0
2.6	[Insert new rows above this one, then hide or delete this row]								
3	**[Task Category]**		Wed 1/08/20	Mon 1/13/20	6		4		
3.1	[Task]		Wed 1/08/20	Wed 1/08/20	1	0%	1	0	1
3.2	[Task]		Thu 1/09/20	Thu 1/09/20	1	0%	1	0	1
3.3	[Task]		Fri 1/10/20	Fri 1/10/20	1	0%	1	0	1
3.4	[Task]		Sat 1/11/20	Sat 1/11/20	1	0%	0	0	1
3.5	[Task]		Mon 1/13/20	Mon 1/13/20	1	0%	1	0	1
3.6	[Insert new rows above this one, then hide or delete this row]								
4	**[Task Category]**		Wed 1/08/20	Sun 1/12/20	5		3		
4.1	[Task]		Wed 1/08/20	Wed 1/08/20	1	0%	1	0	1
4.2	[Task]		Thu 1/09/20	Thu 1/09/20	1	0%	1	0	1
4.3	[Task]		Fri 1/10/20	Fri 1/10/20	1	0%	1	0	1
4.4	[Task]		Sat 1/11/20	Sat 1/11/20	1	0%	0	0	1
4.5	[Task]		Sun 1/12/20	Sun 1/12/20	1	0%	0	0	1
4.6	[Insert new rows above this one, then hide or delete this row]								
5	**[Task Category]**		Wed 1/08/20	Sun 1/12/20	5		3		
5.1	[Task]		Wed 1/08/20	Wed 1/08/20	1	0%	1	0	1
5.2	[Task]		Thu 1/09/20	Thu 1/09/20	1	0%	1	0	1
5.3	[Task]		Fri 1/10/20	Fri 1/10/20	1	0%	1	0	1
5.4	[Task]		Sat 1/11/20	Sat 1/11/20	1	0%	0	0	1
5.5	[Task]		Sun 1/12/20	Sun 1/12/20	1	0%	0	0	1

Creating the main page initial layout.

This will be a draft page where there is a header and an empty body. The header will have a login/logout functionality, a control panel button, and create post button.

Status: completed
Priority: medium
Difficulty: low
Max time: 3 hours
Assigned to **Axel & Wojtek**

Result: *Write here the result of this task.*

Creating an example post.

We will hardcode an example post to have a reference of how it is going to look.
This post must have the following components:

- Title.

- Category.
- Author.
- Content.
- Date.
- Edit button (we will handle when you can edit later).
- Visibility public/draft (we will handle this better later).
- A save button (we will handle this better later).
- Two comments.
  - Author of the comment, one registered user, the other one anonymous.
  - Actual comment content.

Clicking edit will enable the post to be modified. You can actually modify (type).
We will handle the save button with the backend later.

So at this point we will have a completed draft page.

Status: completed
Priority: medium
Difficulty: medium
Max time: 3 hours
Assigned to **Axel & Wojtek**

Result: *Write here the result of this task.*Creating a control panel initial layout.
It is going to be a new page, and it's header will also have a login/logout functionality. Also a return to main page button.

It will have a side panel that will have the following options.

- Your posts
- Your categories

The body will be empty for the moment.

Status: completed
Priority: medium
Difficulty: medium
Max time: 3 hours
Assigned to **Axel & Wojtek**

Result: *Write here the result of this task.*

Creating a category entry example.

This will just represent a category that it will always going to be visible.
The category name will be “new category”.

You can actually edit the category and press enter to modify it.
We will handle the modification in backend later.

Status: completed
Priority: medium
Difficulty: medium
Max time: 2 hours
Assigned to **Axel & Wojtek**

Result: *Write here the result of this task.*

Styling everything.

Making things looking good.
In this part the important thing is that everything has the same theme and a professional feeling.

Status: completed
Priority: low
Difficulty: medium
Max time: 2 hours
Assigned to **Axel & Wojtek**

Result: *Write here the result of this task.*

Designing the database

The database that we are going to use is MySql through the Workbench utility that comes with the bundle.
Once we are familiar with the project, we can clearly see the data entities that we need to store.
In this task we are going to design a diagram representing all the tables and data structure needed in MySql.

We have to store.

- users
- posts
- comments
- categories

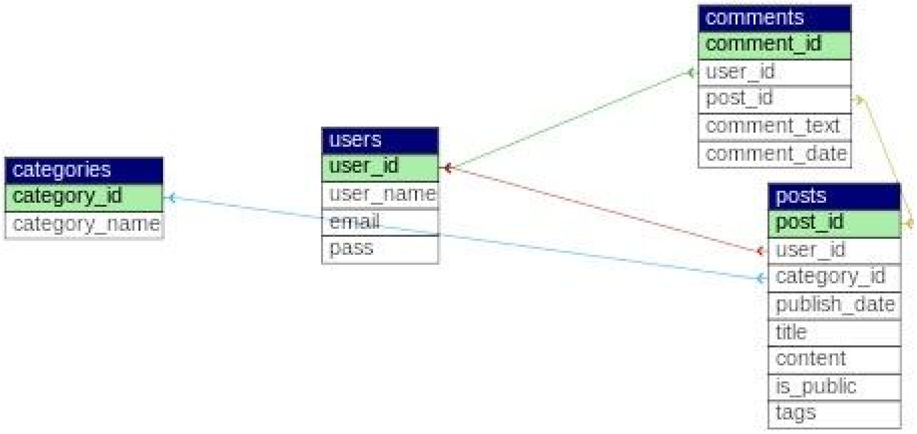
This are the tables. Relations can be clearly seen as well.

A user can have N posts.

A user can have N comments.
A post can have N comments.
A category can have N posts.

Status: completed.
Priority: high
Difficulty: high
Max time: 2 hours
Assigned to **Axel & Wojtek**

Result: *No problems for this task. It took 2 hours a expected to finish. I had to remember some things. Now this sort of tasks can take only 30 minutes to be completed.*



Making PHP adapter file to MySql

We need a way to connect to the previously created MySql database.
We are going to create a database adapter file in php. It will have all the required queries that this project needs.

These are the methods that will be available in the php database adapter file.

Status: completed
Priority: high
Difficulty: high
Max time: 2 hours

Assigned to **Axel & Wojtek**

Result: *Write here the result of this task.*

Making MVC architecture pattern

We create files architecture for organized work and keep code clean.

1. config :
   - config.php
2. controllers :
   - controlpanel.php
   - dashboard.php
   - failure.php
   - login.php
   - register.php
3. libs :
   - app.php
   - controller.php
   - database.php
   - model.php
   - view.php
4. models :
   - entities :
     - category.php
     - comment.php
     - post.php
     - user.php

   - categorymodel.php
   - commentmodel.php
   - postmodel.php
   - usermodel.php
5. other :
   - database.sql
   - populate.sql
   - queries.sql
   - testing.sql
   - testQueries.sql
6. public
   - images/ assets
   - img

    - styles

7. views
    - controlpanel
        - index.php
    - dashboard
        - index.php
    - failure
        - index.php
    - login
        - index.php
    - register
        - index.php

    - category-filter.php
    - footer.php
    - header.php
    - search
    - .htaccess
    - index.php
    - README.md

Status: completed
Priority: high
Difficulty: high

Max time: 5 hours
Assigned to **Axel & Wojtek**

Result: *Write here the result of this task.*

Making config.php

We create config.php for application configuration.

-Database host

```
define('HOST', 'localhost');
```

-Database name

```
define('DB', 'blog');
```

-Database user

```
define('USER', 'root');
```

-Database password

```
define('PASSWORD', "admin");
```

-Database charset

```
define('CHARSET', 'utf8mb4');
```

Status: completed
Priority: high
Difficulty: high

Max time: 2 hours
Assigned to **Axel & Wojtek**

Result: *Write here the result of this task.*

Controller

The controller mediates between the models and views.

Making folder **controllers**

- controlpanel.php

Handle all functions for admin panel. Get all requests from user and send to model or view depend of requests.

- dashboard.php

Handle all functions for main page. Get all requests from user and send to model or view depend of requests.

- login.php

Handle all functions for login page. Get all requests from user and send to model or view depend of requests.

- register.php

Handle all functions for register page. Get all requests from user and send to model or view depend of requests.

Status: completed

Priority: high
Difficulty: high
Max time: 4 hours
Assigned to **Axel & Wojtek**

Result: *Write here the result of this task.*

Making folder **libs**

- app.php

File app.php is a first thing executed in our application.
Handling routing that it will be redirect to proper controller.
If controller specified does not exists, it will redirect to dashboard by default.

- database.php

Prepare a connection to mySql.

- controller.php

Parent class for controller.

- model

Parent class for controller.

- view

Parent class for controller.

Status: completed
Priority: high
Difficulty: high
Max time: 6 hours
Assigned to **Axel & Wojtek**

Result: *Write here the result of this task.*

Making folder **models**

Folder models handle from entities, it send queries to database and response to controller.

The entities folder contains all classes.

entities/

- category.php
- comment.php
- post.php

        - user.php

- categorymodel.php

Handle all functions about categories.

- commentmodel.php

Handle all functions about comments.

- postmodel.php

Handle all functions about posts.

- usermodel.php

Handle all functions about users.

Status: completed
Priority: high
Difficulty: high
Max time: 6 hours
Assigned to **Axel & Wojtek**

Result: *Write here the result of this task.*

Making folder **views**

This part deals with presenting the data to the user. This is usually in form of HTML pages.

controlpanel/

- index.php

Presenting data about panel Admin.

dashboard/

- index.php

Presenting data about main page.

login/

- index.php

Presenting data about login.

register/

- index.php

Presenting data about register.

- category-filter.php

Presenting data after when you choice category will be filtering all posts and display by category choice.

- footer.php

Presenting footer.

- header.php

Presenting footer.

- search.php

Presenting data by search filter.

Status: completed
Priority: high
Difficulty: high
Max time: 5 hours
Assigned to **Axel & Wojtek**

Result: *Write here the result of this task.*

Making folder **other**

In this folder we keep our files mySql what we was used to this project.

Status: completed
Priority: low
Difficulty: high
Max time: 3 hours
Assigned to **Axel**

Result: *Write here the result of this task.*

Making folder **public**

This folder contains everything about styles CSS and Images.

Status: completed
Priority: low
Difficulty: low
Max time: 1 hours
Assigned to **Axel & Wojtek**

Result: *Write here the result of this task.*

Making other **files**

- .htaccess

Configuration file for use on web servers running the Apache Web Server software.

- index.php

Requires everything that the application needs from libs, then is calling for app.

- README.md

A README is often the first item a visitor will see when visiting your repository. README files typically include information on:

- What the project does
- Why the project is useful
- How users can get started with the project
- Where users can get help with your project
- Who maintains and contributes to the project

Status: completed
Priority: high
Difficulty: high
Max time: 3 hours
Assigned to **Axel & Wojtek**

Result: *Write here the result of this task.*