CSC 648/848 Software Engineering – Spring 2017

Milestone 0

02/09/17

Anthony John Souza, SFSU

Dragutin Petkovic, SFSU

Rainer Todtenhoefer, Fulda University, Germany

1.Introduction

The purpose of this milestone is to help you set up the development infrastructure that your team will be using this semester to develop its final presentation website. This infrastructure consists of: a) *deployment server and tools/stack* for delivering the final running application; and b) *development server and tools* for development and testing of the code. Both of these are running on the same AWS server set up by instructors.

This environment and setup is close to what you will be required to know and do at work, so M0 is a great experience to improve your marketable skills. While all work is to be done individually, you are encouraged to work together as a group to complete the task. **This means that you can help each other, ask for help, work in pairs and are in fact encouraged to do this to not only help with M0 but also build the teamwork. But ultimately, each team member has to do it him/herself and learn the tools used for final team project.**

M0 carries 10 grade points (not 5 points as before). Information about grading can be found at the end of the document.

Your first task is to read through the entire document. Then, follow the steps in the setup Information (section 2) to setup your access to the various accounts and services this infrastructure provides and to install the necessary tools. You will also give access to instructors so they can monitor your work.
In section 3 we then describe actual M0 task where you are to create a joint web page. This WWW page can be used in your final project too.

This is a brand new infrastructure for the CSC 648/848 class, designed to give teams more freedom and responsibility in setup. Hence, there may be problems. Please contact ajsouza@mail.sfsu.edu if you encounter any problems that you cannot solve yourself with your team.

Finally, M0 in itself is a great vehicle to help you learn a very important part of SW engineering work which is setup, use and maintenance of the development and deployment infrastructure.

Note that we use *group* or *team* interchangeably.

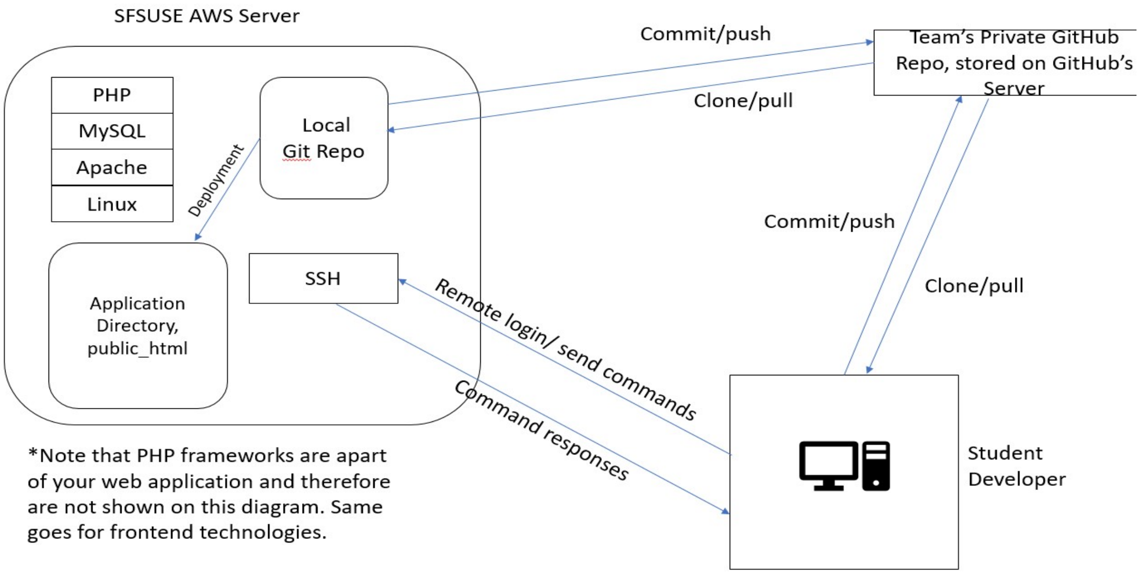
2.Overview and installation of the SW environment, frameworks and tools

The class development AND deployment infrastructure comprises:

On the sfsuse.com server:

- A Linux shell account for each student
- A Linux shell group account shared by all members of a team
- A MySQL database and account for each student
- A MySQL database and account shared by all members of a team
- LAMP stack with which to develop your web app
- Git, used for version control
- **One of five preapproved PHP Frameworks for app development (team will choose one)**

Overview of the system deployment is in Fig 1. Items A through F will be installed by class CTO (Anthony). Item G will be installed by each group's CTO. Sfsuse.com server will be provisioned and maintained by Anthony, class CTO, running on AWS (Amazon Cloud).



*Note that PHP frameworks are apart of your web application and therefore are not shown on this diagram. Same goes for frontend technologies.

Fig. 1 – overview of the class infrastructure

Note that for most tools and services there will be: a) *individual student accounts* used individually by students for the development and testing; and b) *team (group) account*, private for each team, with access by all team members (and instructors) where the work of all team student members (code, documents) will be merged and the application deployed. **All milestones and demos and final application must be delivered and deployed from team accounts.**

2.1 Individual student Linux Shell Account

Each student has a Linux shell account on the sfsuse.com server setup by the class CTO. Due to security concerns, you are only allowed to log into this account using SSH. There is no FTP service available. The purpose of the individual shell account is to allow students to do fast prototyping, learn new technologies, and run sample code without effecting their groups project or having to run their own environment.

Linux shell accounts may not receive emails, and while Linux shell accounts may send emails, you should limit sending emails only to your SFSU email account. **Abuse of email services on the server will result in termination of your account on the server and potentially a grade of 0 for the product portion of your grade.**

Individual Shell Accounts

Each Linux shell account has a public_html directory. Files in this directory are available by the web at the URL http://sfsuse.com/~*username*, where *username* is your Linux shell account username.

Required Software for SSH Access

Mac and Linux Users

Mac and Linux users already have the software necessary to login, the ssh command from a terminal prompt. Since you would have to point ssh to your key every time you ran it, it is recommended that you modify your ~/.ssh/config file to include an alias for the sfsuse.com server. Figure 1 shows a sample configuration entry.

```
Host sfsuse
        User username
        HostName sfsuse.com
        IdentityFile ~/.ssh/csc648848su16.pem
```

Figure 1. Sample .config file entry

Windows Users

Windows users need to install software which allows them to communicate via SSH. The recommended software for this is PuTTY, available at http://www.putty.org. You will need to configure this to work with SSH keys.

Logging into your account

Username

Your username for this account is the part of your mail.sfsu.edu email address before the @ sign. For example, a user with the email address jdoh@mail.sfsu.edu would have the username jdoh.

Password

There is no password for your individual shell account. You must use SSH keys to access your account. Example key is given below.

SSH Keys

You will be receiving an email containing a private key. At the bottom of the email, there is a section looking something like the example in
Figure 2.

```
-----BEGIN RSA PRIVATE KEY-----
kuy/WyqFoeX/15d+GKp9cQKBgGdgnbiUBFQoHUMxCYbIpxNbZlP0dECDAgEOyv2S
+TPc/0vTqUjJ9UOKtGHXDSq3j4tNvWPnCwKvLC+QBo4jgpwIYuzFGWAmDJmTF3e
w
6T+ZA7IpFJBSkYVhdoEd4Vdz77/k89cbcpUZVwlXdQFs8PsEl10SPT698UKfP8g0
j4WxAoGAV4qXWKLgy2C2JowWc4nVnb/XLGxC86LPZQWfm7Gil8wZzit5C9XjU3zX
XK9gQm6C/VsuUmEFUScXcVR/B9pHo1mcRwSQUgIDyyHFkfgXXQm3HhJAH59
+tBTh
u1yBHoS25oaEvaiin1s9uTabtVtRcAYPmaVbOueuYTf71krPBq8=
-----END RSA PRIVATE KEY-----
```

Figure 2. Example private key

In order to create a key, you must cut and paste this section of the email into a text file, including the lines containing BEGIN and END RSA KEY with all the "-
" characters with NO blank space before or after. It is suggested that you name the file csc648848su16.pem, since this key will be used for more than just logging into your Linux shell account.

Mac and Linux Users
For Mac and Linux users, you should place this file in the ~/.ssh directory, and chmod permissions to 400.

Windows Users
Windows users need to point PuTTY to this file when configuring ssh keys.

2.2 Team (Group) Linux Shell Account

Your team shell account is shared by all members of the team. The purpose of the group shell account is to serve your team's application to the web. Also code review and grading will ***ONLY*** be done from this group shell account. It is recommended that your team choose one team member to be the person responsible for the account, someone preferably with some Linux experience.

The username of your group account is **sp17g*nn*** where *nn* is the number of your team (with one leading 0 for single digit team numbers).

The private key file for individual shell account will also work for you group shell account. Simply change the username on your ssh command. For example:

```
ssh -i <path to key file> sp17gnn@sfsuse.com
```

Important!
The public_html directory in this account is from where your group must serve its final project website.

2.3 Individual MySQL Database Account

The sfsuse.com server is running MySQL for its database server. Each student has had setup for them an individual account on this database, paired with an individual database. The class CTO has set these up for you. As a student you are allowed to modify your own database as you see fit, but not create new databases. The purpose for the individual database account is to allow students to prototype, test, and learn to use MySQL without affecting the teams shared production database based

There is no GUI interface provided to the database, and for security purposes the database is not accessible via an *external* port on the server. This means that for database management you can either log into your Linux shell account and use the command line interface to MySQL, or you can use a GUI tool as described in Recommended Software below.

MySQL Username and Password

Your username is the same as your Linux shell account. Your database is named student_*username* where *username* is your Linux shell account / MySQL account username. The default password for this account is your student ID. You must change this password immediately. To do this, start MySQL with the command `mysql –u username –p`, where username is your MySQL username.

From the mysql command prompt, type the following two commands:

```
SET PASSWORD FOR 'username'@'localhost' = PASSWORD('password');
FLUSH PRIVILEGES;
```

where *username* is your MySQL username, and *password* is your new password. If you forget your password, you will have to contact the system administrator to reset it for you.

The easiest GUI to get started with is MySQL Workbench, available at https://www.mysql.com/products/workbench/. This is expecting to be able to connect to the port 3306 on the server with the MySQL server running. As mentioned before, this port is not accessible remotely.

Connecting to MySQL Server via command line or via MySQL Workbench

There are two ways you may access your individual or group database. One way is logging into the remote AWS server and accessing your database via command line with the following commands:

```
mysql -u username -p
```

(where username is your database username and password is your db password)

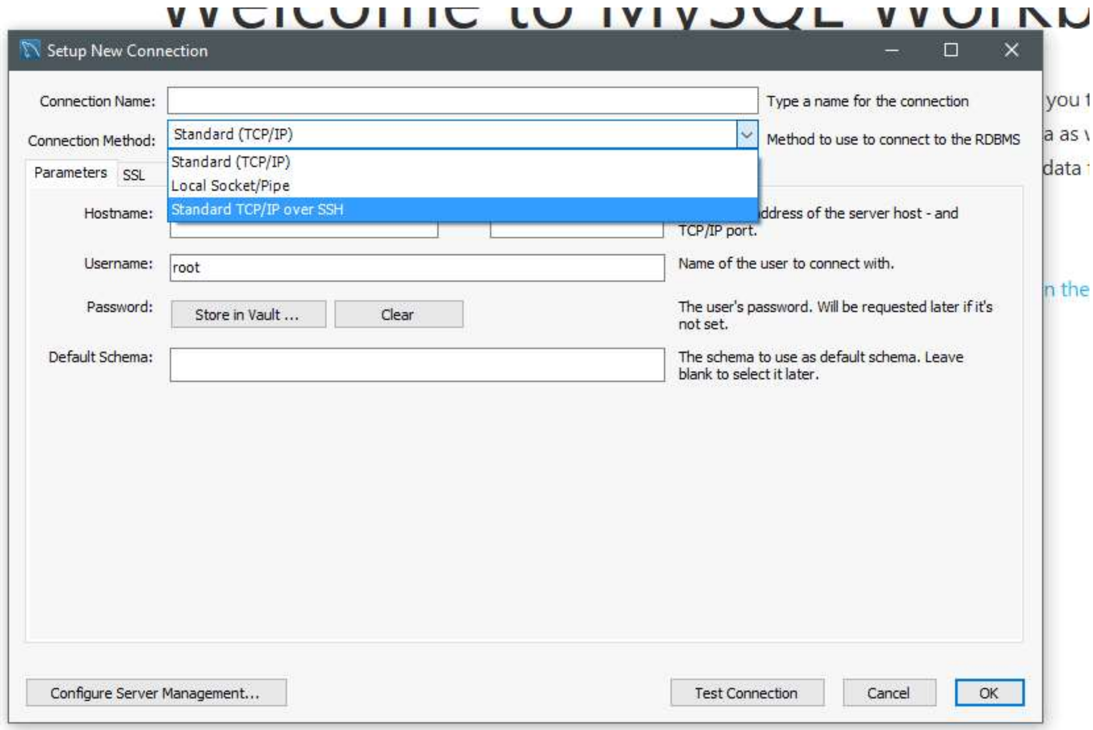
```
connect db_name
```

(where db name is the database name you wish to connect to)

At this point you may start executing SQL commands against your connected database.

The other way is via MySQL Workbench. This is GUI application used to manage MySQL databases.

To connect to your database you first need to create a new connection. This can be done by clicking the plus sign(+) near the text MySQL Connections. This will open a new tab. You will then select a connection type. The connection to select is “Standard TCP/IP over SSH”. Figure shown below.



Setup New Connection

Connection Name: — Type a name for the connection

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

Standard (TCP/IP)
Local Socket/Pipe
Standard TCP/IP over SSH

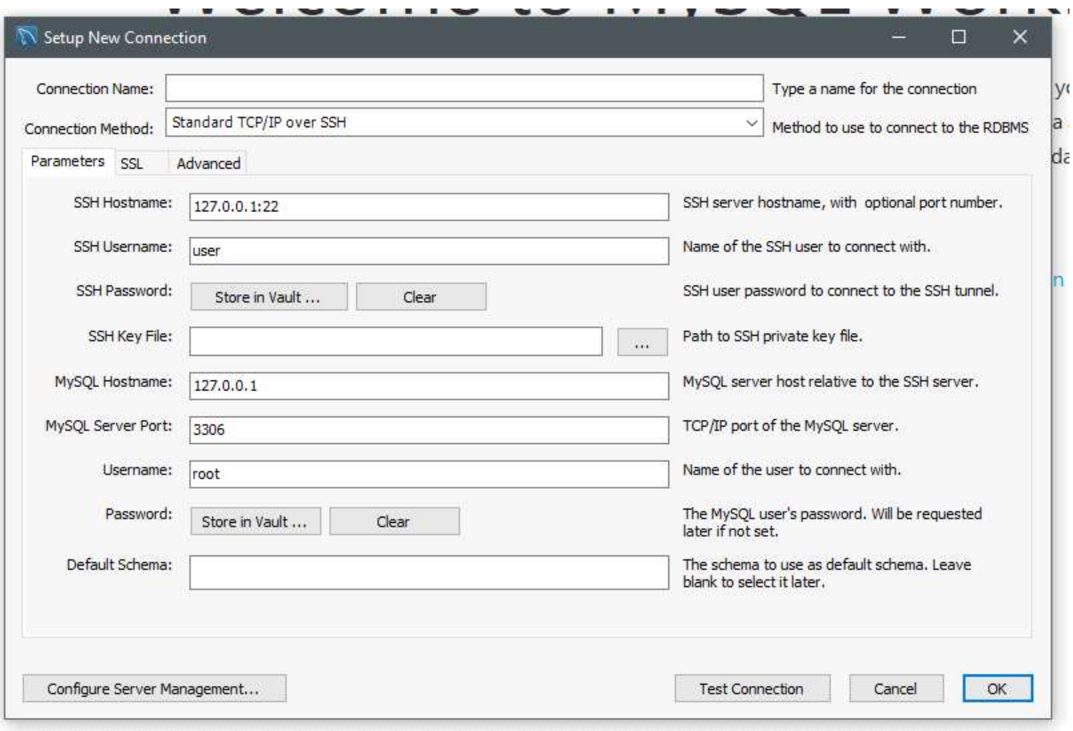
Hostname: — ddress of the server host - and TCP/IP port.

Username: root — Name of the user to connect with.

Password: — The user's password. Will be requested later if it's not set.

Default Schema: — The schema to use as default schema. Leave blank to select it later.

Next, you will need to fill in the required information for creating a new connection over SSH.



Setup New Connection

Connection Name:		Type a name for the connection
Connection Method:	Standard TCP/IP over SSH	Method to use to connect to the RDBMS

Parameters | SSL | Advanced

SSH Hostname:	127.0.0.1:22	SSH server hostname, with optional port number.
SSH Username:	user	Name of the SSH user to connect with.
SSH Password:	Store in Vault ... Clear	SSH user password to connect to the SSH tunnel.
SSH Key File:		Path to SSH private key file.
MySQL Hostname:	127.0.0.1	MySQL server host relative to the SSH server.
MySQL Server Port:	3306	TCP/IP port of the MySQL server.
Username:	root	Name of the user to connect with.
Password:	Store in Vault ... Clear	The MySQL user's password. Will be requested later if not set.
Default Schema:		The schema to use as default schema. Leave blank to select it later.

Using the above figure as an example, fill in the following information:

- Connection Name: sfsuse_db
- SSH Hostname: sfsuse.com
- SSH username: your_ssh_username
- SSH Key File: path you your key file
- MySQL Hostname: 127.0.0.1
- MySQL Server Port: 3306
- Username: database username, same as shell account name
- Password: database password, should be student id, unless changed.
- Default Schema: student_username, this your database name, where username is your database account name.

2.4 Team (Group) MySQL Database Account

As with the group Linux shell accounts, your group has its own MySQL database account and database. The purpose of the group MySQL account is to store and serve data to your team's project. The username for this database account is the same as the group's Linux shell account username.

The database for the group is the group's username. No student_ *username* is pre-appended to the database name as with individual MySQL accounts. The default password for the team's MySQL account is csc648sp17gnn, where nn is your group number.

Important!
The group database is the database that must be used to serve data for the team's final project.

2.5 LAMP Stack and other preapproved tools

A standard LAMP stack has been provided for your development enjoyment. The purpose of the LAMP Stack is to provide teams in CSC648 with a basic web development and deployment infrastructure. You are only given the bare minimum to run a WWW website.

SFSUSE Server Information

- **Host: Amazon Cloud**
- **OS: Ubuntu Server, Version: 16.04**
- **MySQL Version: 5.7**
- **PHP Version: 7.0.13**
- **OpenSSH Version: 7.2**
- **Git Version: 2.7.4**
- **Python: 2.7**
- **Ruby: 2.3.1**
- **nodejs: 4.2.6**
- **npm: 3.5.2**
- **Less: 481**
- **Sass: 3.4.23**

If you do not already know how to develop in PHP, please refer to abundant online resources. While not required for submitting **M0**, in addition to learning PHP, there are going to be front end GUI technologies that will need to be installed. ***There will be no general instructions given for installing these technologies as it may differ depending on the PHP framework you choose.*** But to give a list of the few popular, useful and preapproved frontend GUI technologies:

- JavaScript (installed by teams)
- jQuery (installed by teams)
- Bootstrap (installed by teams)
- CSS (installed by teams)
- SASS (CSS scripting language, installed by class CTO)
- LESS (CSS preprocessor, install by class CTO)

Please keep the following in mind:

- None of the above technologies are required for submitting M0.
- You do not need to use ALL these technologies for your team's project.
- As a team you should decide which ones you would like to use and will report it in M1.
- When ready they will need to be installed to your groups Shell account.
  - They may be installed to your individual shell account only for testing and learning purposes. Class CTO and CEO will never grade from individual shell accounts.

2.6 PHP Frameworks

Choosing a Framework

For this class, you will be allowed to choose which PHP framework to use for development of the team project. Not all frameworks are created equal and to this point you will not be allowed to use any Framework. We want to avoid frameworks that are too high level, since they are very restrictive in what you can do with them and also prevent you from learning details necessary for CS majors.

Below is a list of preapproved PHP Frameworks:

- Symfony
- CodeIgniter
- CakePHP
- ZendFramework
- Slim microframework

As a group you need to decide which single PHP framework you would like to use. This includes willingness and ability of the whole team to use and learn the tool so this needs to be verified by team lead and team CTO. Any other framework has to be requested and then approved in writing (e-mail) by class CTO Anthony.

Once a PHP framework has been chosen, follow the steps to install the framework to your individual Shell account as well as installing the framework to your groups shell account. Please note that installing the framework to the group shell account should be done by one team member and should only happen once. Make sure to follow the configuration steps as well to hook up your project to the provided MySQL database. If there are any issues or questions, please email the class CTO.

Team lead and team CTO need to ensure that ALL team members learn the tools by for example organizing a training session.

2.7 GitHub Team (Private) Repo

The purpose of this part of the exercise is for your team to set up the team private GitHub repository that is going to be used for storing your team's project and be accessible only to approved team members (and instructors). Only one member needs to set up the private repository.

(As a first step, all team members have to have their own GitHub account. This is mandatory, NO EXCEPTIONS - see step 1 below)

*Setting up Private GitHub Repo Via Student **Free** Development Pack*

One member of your group will be designated as GitHub admin. It is going to be his/her responsibility to setup and maintain the private repository for your teams' project. Only one team member needs to apply for a free Student Development Pack, but all are encouraged to do so if they want. There are many added benefits and free coupons that come with GitHub's student developer pack. **This developers pack is free; the benefits you get are free for a certain amount of time.**

If a member already has the ability to create private repos on their GitHub account, this is sufficient and you can skip applying for a developer's pack.

Applying for <u>free</u> GitHub Student Developer Pack.

- If needed, create a GitHub account. If you already have an account you can skip this step.

    - When creating the GitHub account, select that you will have public repos. DO NOT SELECT private repos, or you will be asked to enter credit card information.
- After the GitHub account has been created navigate to https://education.github.com/pack and apply for a Student Developer pack. You will be asked to log into you GitHub Account. It takes at most 1 to 2 days to be approved, but it happens usually within hours.
- After your Student Developer Pack has been approved you can navigate back to the same link from step two. Click "Get My Pack", and again will be asked to login into your GitHub account. But this time after logging in the page will be different and you will be able to use the listed benefits on the page.

Once your pack is approved your groups GitHub admin will be required to complete the following steps:

- Create a team (private) repo for your team's project
- Add ALL team members to private repo
- Add BOTH instructors to your team's private repo.
    - CTO Username: asouza88
    - Instr Username: csc648848Instr
- You will later push the initial installation of your teams chosen PHP framework to the private repo (see more on this below).

- **M0 task**

Step 1: Selecting and Deploying PHP Framework

The purpose of this task is for you as a team to select one of the following approved PHP frameworks. The frameworks are relisted below.

- Symfony
- CodeIgniter
- CakePHP
- ZendFramework
- Slim microframework

When deciding on which framework to use you should keep several key factors in mind. Some key factors may be:

- How often the framework is updated

- How well supported is it
- How mature the framework is
- How well the documentation is
- How good is the community
- What features do is offer compare to other frameworks
- How easy is it to use/learn
- And finally: how effectively will team members be able to learn/use the framework

There are more, but a small list is given above. Note that this decision is the team decision and selection of frameworks does not influence the grade (e.g. you can choose the simplest one).

After your team has chosen a framework, the team CTO should install the framework and make the initial push to your team's private GitHub repository. After this, every team member has to clone the team's private GitHub repository and proceed to the next task.

Step 2: Create a Team Website

The purpose of this part of the exercise is to get you to work individually to create a webpage within your team's framework context, and then for your team to join these pages together using your teams GitHub account into a single site. We recommend that you in fact create ABOUT WWW page which introduces the team members. This can then be part of your final application and is great for your portfolio.

Each individual in your team should clone the team's private GitHub repository into his/her individual shell account or onto your local computer, and within the framework create a page that at a minimum displays their name and their or some other image (if you are comfortable it is good idea to use your own image which is useful for your ABOUT page and portfolio – this is of course optional). Please make sure that you have the right to use that image. This work has to be completed by individual student and then pushed to the team's private GitHub repository. The file(s) created/added should then be merged into the team's private GitHub repository. The website must be deployed onto the team's group shell account.

One of your teammates will need to modify your framework's home page to point to all the different team member pages. The finished site must look something reasonably close to the example in . The user 1...user 6 buttons should be replaced with buttons pointing to the individual pages labeled with the username of the person who created that page. The website shall be deployed onto the team's group shell account.

Figure 3. Example mockup of Milestone 0 team home page.

This single site must **be served from the public_html directory in your team's *group* Linux shell account.** This is the same way you will deploy your final app, hence this is useful to make sure you learn and test it.

4. Submission and Grading of M0

Submission of M0 for grading

All projects will be inspected and graded on the due date. The key for grading will be your team WWW page served from **the public_html directory in your team's *group* Linux shell account.** Individual accounts will also be checked. Emphasis is on correctness (not so much on UI design of the final team page) and proper usage of development tools. There is no need for any additional submissions.

Grading of M0

This assignment is worth 10% of your class grade, as indicated in the syllabus. The grade for this assignment will be determined according to the following criteria:

Category	Points
Correct use of Git	4
Correct team WWW page functionality, deployment and proper usage of team's framework for creating web page	6
Total:	10