



TOSHKENT SHAHRIDAGI INHA UNIVERSITETI INHA UNIVERSITY IN TASHKENT

Module Name: Internet Programming
Module Code: SOC2110
Project Type: Group Project (4-6 students)
Date Set: 23.03.2020
Deadline: 10.05.2020 (midnight)
Weighting: 20%

Instructor: Dr. Sarvar Abdullaev
Email: s.abdullaev@inha.uz
Office Hours: Wed, 14:00 - 15:00
Fri, 15:30 - 16:30
My Zoom ID: <https://us04web.zoom.us/j/5054265525> (I will be available for your questions during office hours)¹
Office: Room B411,

Learning Outcomes

1. Understand the functional requirements of chosen business scenario
2. Collaborate on a web project using Github
3. Design GUI using HTML and CSS for chosen business scenario
4. Design database schema based on the requirements of chosen business scenario
5. Implement back-end functionality using PHP and relevant frameworks
6. Implement front-end functionality using JavaScript and relevant frameworks
7. Integrate with other services available in web
8. Deploy a completed web application to a public hosting

Project Deliverables:

1. A **PRINTED REPORT** as specified in this assignment²
2. **ALL SOURCE CODE** of your web application in corresponding GitHub Classroom Repository.
3. **ALL SOURCE CODE** and **REPORT** of your web application in eClass.

¹ During the quarantine period, you can talk about your project or ask questions using [Zoom](#). Please, connect during office hours. If you would like to talk to me outside my office hours, send an email and confirm the timing with me beforehand.

² If the quarantine does not finish until the deadline, printed report is not necessary.

Design Project Overview

You are required to develop a web application which involves both client-side and server-side programming. It should also use database. You should upload your final source code to corresponding GitHub Classroom repository and host your application on a remote server. Web application should be developed using HTML, CSS, PHP, MySQL, JavaScript and other related technologies covered in lectures. You must choose one of the below business scenarios for your web application. If you have a different project idea, please write an email or contact me in my office hours to verify.

Business Scenarios:

You have been employed by company X to develop a corporate website. You can select the area of business of the company X from given list below, or come up with your own idea.

- Hotel – customers can book rooms beforehand for certain period of time.
- Restaurant – customers can submit orders for food in menu and books tables.
- Publisher – subscribers can download books, authors can submit proposals to publish
- Taxi/Transfer – customers can book taxi beforehand indicating pick up and drop off locations.
- Online store – customers can buy goods online by adding items to their shopping cart and checkout at the end.
- Announcements board – users can post their announcements under different categories, moderators can modify, reorder or remove announcements.
- Personal blog – author can post blogs and subscribers can view and comment those blogs. As services, web site can advertise the skills and expertise of the author of the blog.
- Online surveys/tests – author can create surveys and tests, users can submit their answers and view results. As services, web site can advertise the functionality of taking customize online tests on given subjects.

The functionality of a web application that you develop for above business scenario must comply with the requirements lists in evaluation criteria.

How to Submit Your Project?

1. Accept Github Classroom Group Assignment via invitation link shared in Telegram channel
2. Create your own team OR choose existing team you would like to join.
3. Github Classroom will create a project repository. Each team member has to clone it to his/her computer.
4. Develop your project according to the requirements specified in next section. Collaborate with your team mates effectively on Github³.
5. Prepare a project report as specified in next section and include it into your project's main folder.
6. Upload your project to public hosting.
7. Push final version of your project along with the report to its Github repository.
8. Zip your project folder which contains your project source code and project report, and upload it to eClass before the deadline. Note, you must not include third-party libraries (e.g. *node_modules*, *vendor* folders) into your zipped project folder.

³ Additional session will be made on Github collaboration.

Evaluation Criteria:

Section 1: Web Application

[weight 80%]

You are required to implement a web application for some company of your own choice using all the materials covered in lectures. Your application should support following general functionalities:

1. Content (weight 20%)

This rubric evaluates your general skills in designing simple HTML web pages. In order to demonstrate it, you should create:

- **Company Landing Page (index page):** this page should contain short information about the company along with its logo and some relevant photos.
- **Products/Services Page:** this page should contain information about products/services this company provides. It should also allow searching particular products/services based on some parameters.
- **Registration and "Sign In" Pages:** this page should collect important information about its potential customers and allow them to sign in to their personal profile.
- **Make Order/Booking/Blog Post Page:** this page should create a record in the database which can be viewed by site administration. The administration can fulfil or cancel the order, and this information should be updated in database. Customer has to receive an email notification about the order status change. Depending on your business case, the statuses can be different too.
- **Process Order/Booking/Blog Post Page:** this page must be visible only to Admin user. In this page, company workers can process submitted orders by assigning fulfilled or cancelled statuses to them. Depending on your business case, the statuses can be different too.
- **Contact Us Page:** this page should provide a simple form for leaving feedback about the web site, services/products to the administration. It should send an email to corresponding address when feedback is submitted.

2. Look and Feel (weight 10%)

This rubric evaluates your CSS skills for designing the format and the layout of your web page. You can write the entire stylesheet yourself, or use existing stylesheets such as Bootstrap. You should customize the default stylesheet to create your company's unique identity. The layout of your choice should be intuitive and commonly used by other well-known web sites. You can also include new CSS3 features such as Canvas and SVG for vector graphics, Google Fonts for better typesetting, Media Queries for device-specific formatting, transformations and animations for better interactivity, etc. You may receive higher points if you use Bootstrap or any other well-known stylesheet in your web site. **Note**, you are not allowed to use ready-made HTML templates from web though, and create your web page yourself.

3. Forms and Input Validation (weight 10%)

This rubric evaluates your skills in using forms for collecting user input. All data submitted via forms should be validated either on client-side or on server-side (or both) using Regular

Expressions and corresponding error message should be shown to the user. At least one of your forms must contain following fields:

- Username which must contain at least 5 alphanumeric characters
- Password which must contain at least 5 word characters (alphanumeric, %,\$, -, _, etc)
- Confirm password which should be the same as the password
- Email which must match the standard email format
- Phone number which must match the standard phone number format (e.g. +998-99-1234567)
- Date of birth which must match the standard date format (e.g. 23-03-1993)
- Postal code which must contain exactly 7 digits (e.g. 100011)
- City name which must contain words only
- Passport number which match the standard passport number (e.g. AA1234567)

4. Database (weight 10%)

This rubric evaluates your skills in storing and retrieving data from a database. Your web site should store all user input collected via forms in a database deployed to a MySQL server (other DBMS such as Postgres, SQLite, etc are not allowed). You may use PHP's *PDO objects* to make a database connection and *PDOStatements* to interact with the database via prepared SQL statements. Make sure that you implement a repository for each entity as a **data access layer (DAL)** of your web application. If you use Laravel's Eloquent, then define a repository for accessing ORM objects. Also your database schema should be migrated and seeded into any database using Laravel's migration tools. Your database schema should contain at least **5 tables** which are related to each other through Primary Key/Foreign Key relationship. Make sure that you also have a dedicated *Users* table for implementing authentication and authorization features.

5. Server-side Framework - Laravel (weight 10%)

This rubric evaluates your skills of using Laravel for building enterprise level web applications. Your web application should use Blade for rendering templates, routing for mapping URLs to controller actions, Eloquent for ORM and database connectivity, migrations for defining database schema, middleware for authentication/authorization, etc. To sum up, the project has to make full use of Laravel's key features. If you DO NOT use Laravel, then you should implement the entire web application using plain PHP and you will ONLY lose mark for this rubric (i.e. 10% of total mark). You are not allowed to use other server-side frameworks.

6. Client-side Scripting (weight 10%)

This rubric evaluates your client-side scripting skills using JavaScript. This may involve adding extra interactivity to the front-end such as client-side input validation, pop-up windows and richer UI through existing components such as Bootstrap. You can also use Angular for building single-page UI, but it is not necessary for this project. Use of tools such as Webpack and Gulp for running tasks, NPM for managing dependencies, Typescript and Sass for transpiling source code into browser-compatible JavaScript and CSS or using other extensions for minification, uglification and concatenation are also optional.

You are encouraged to integrate external services into your web site through jQuery or other AJAX tools (e.g. HttpClient for Angular). You should find an external web service that is useful for the web application you are developing. You can use any of the open APIs from following directories and curated lists:

- <https://www.programmableweb.com/apis/directory>
- <https://github.com/abhishekbantia/Public-APIs>
- <https://github.com/toddmotto/public-apis>

7. GitHub Activity and Hosting (weight 10%)

This rubric evaluates your skills in collaborating with your teammates using tools such as Github and deploying your web application to a public hosting. This involves working with Git branches for feature development, creating pull requests in Github and merging branches into master branch.

You should also upload and configure your web application to a public hosting. This involves working with cPanel for setting up FTP server and MySQL database. There are a lot free PHP hosting providers available in web. Some of them are listed below:

- <https://www.000webhost.com>
- <https://devcenter.heroku.com/articles/getting-started-with-php>
- <https://titanichost.net/free-hosting/>
- <https://infinityfree.net>
- <https://www.freehostia.com/>

Section 2: Report

[weight 20%]

Your report should contain following sections:

- On the cover page, the name of the company and its short motto. Names of all team members and their IDs, **LINK TO YOUR GITHUB REPO** and **LINK TO YOUR WEB SITE**.
- Brief description of the company and its key business (non-business) activities. Short information about its customers and workers, and their corresponding roles.
- Screenshots for each feature/requirements for the web site. Below each screenshot, you should briefly describe what functionality given screenshot realizes and how it has been implemented.
- Summary of each member's contribution to the project.
- Logins and passwords for different users of your web site such as customer, admin, etc.

What is forbidden?

- You are NOT ALLOWED to use ready-made HTML templates, CSS animations/graphics or JavaScript/PHP code snippets taken from some other projects else in your project.
- You are NOT ALLOWED to use ready-made Content Management Systems (CMS) such as Wordpress, Joomla, Drupal, etc in your project
- You are NOT ALLOWED to use any third-party back-end framework other than Laravel in your project.
- You are NOT ALLOWED to use any third-party front-end framework other than Angular.

Failure to comply with above rules may result in a lower or even a zero mark for this project.

What is allowed?

- You are ALLOWED to use third-party components (NOT frameworks) for client-side programming such as jQuery, lodash, Angular Material Design, PrimeNg, Bootstrap, etc, BUT their source code must be stored separately from your own code. It is OK to use dependency management tools such as *npm*, *composer*, *bower*, *yarn*, etc to install such components in your project folder. This way they will be stored separately in a folder like “*node_modules*” or “*vendor*” and do not get mixed with your own code.
- You are ALLOWED to use certain hacks, workarounds or tricks written in public forums such as *StackOverflow*, *StackExchange*, *CodePen*, *jsFiddle*, *Github*, etc, BUT you should put a comment with a URL above the code snippet that you copy/pasted from a public forum.

Online Learning Materials:

Git and Github:

You can learn more about using Git and GitHub from an interactive tutorials given below:

- <https://www.codecademy.com/learn/learn-git>
- <https://try.github.io/levels/1/challenges/1>
- Branching/Merging: <http://learngitbranching.js.org>

Laravel:

You can learn more about using Laravel from tutorials and video lectures below:

- <https://www.tutorialspoint.com/laravel/index.htm>
- Documentation: <https://laravel.com/docs/5.6>
- Video Tutorials for Some Projects on Laravel in LaraAcademy:
 - <https://laracademy.co/courses/laravel-tidbits>
 - <https://laracademy.co/courses/build-a-basic-blog-from-scratch>
 - <https://laracademy.co/courses/emails-with-laravel>
 - <https://laracademy.co/courses/how-to-make-a-landing-page>
- Laravel Tutorials on Youtube:
 - <https://www.youtube.com/watch?v=EU7PRmCpx-0&list=PLl1GF-RfqBYhQsN5WMXy6VsDMKGadrJ->
- Lessons in LaraCasts (some lessons are not free):
 - <https://laracasts.com/lessons>
- Laravel Awesome List (curated list of tools and learning materials):
 - <https://github.com/chiraggude/awesome-laravel>

Angular:

- Getting Started with Angular: <https://angular.io/start>
- Angular Official Documentation: <https://angular.io/docs>
- Angular in Russian: <https://webdraftt.com/tutorial>
- Angular Course on Udemy:

- <https://www.udemy.com/course/the-complete-guide-to-angular-2/>
- Free Video Tutorials on Youtube:
 - <https://www.youtube.com/watch?v=0eWrpsCLMJQ&list=PLC3y8-rFHvwhBRAGFinJR8KHlrCdTkZcZ>
 - <https://www.youtube.com/watch?v=YN8zNnV0sK8>
 - <https://www.youtube.com/watch?v=K5FL6-6Wmjw>
- Angular Components:
 - Material Design: <https://material.angular.io/guide/getting-started>
 - PrimeNg: <https://www.primefaces.org/primeng/>
 - Bootstrap: <https://ng-bootstrap.github.io/>
- Angular's Awesome Curated Lists:
 - <https://github.com/PatrickJS/awesome-angular>
 - <https://github.com/brillout/awesome-angular-components>

Disclaimer:

- I reserve the right to grade each member of the team separately if I have a suspicion on his/her contribution to the project. I will review the history of commits in your project's Github repository, and if I find a member who has not contributed at all to the project, I will call this member for a short viva regarding his/her contribution. If member fails to show up, he/she will receive 0 for the project.
- I reserve the right to call the whole team for a short viva if I have suspicion about the quality of work submitted. If team fails to explain how web application has been implemented, each member of the team receive 0 for the project.