COM S/SE 319: Construction of User Interfaces

Spring 2024

Final Project

[Total Points: 100]

Assignment published: Monday April 15th, 2024

Assignment Due: Friday, May 9th, 2024, 2:00PM

1.- Overview:

The Final Project serves as evidence that the student has comprehensively grasped all the topics covered throughout the course, demonstrating practical application of acquired knowledge. It necessitates the utilization of various web technologies, including HTML, CSS, JavaScript, React, NodeJs, Express, and either MongoDB or any SQL database, to solve a self-selected problem. Optionally, the student will include Bootstrap or Tailwind to improve CSS styles.

The problematic to solve and the information contained on the website is the preference of the students.

The Final Project is composed of the next elements :

- Final Project :

o	Proposal	:	5 %	already submitted
o	Software	:	20 %	Week 16
o	Documentation	:	5 %	"
o	Video	:	5 %	"
o	Demo	:	5 %	"

The previous work, `Proposal` has already been worked on by the students, handed it in previous dates and they are not required in this document.

Every component has its own value for the final grade. Therefore, each can be reviewed and graded over 100.

2.- Software:

The `Software` of the Final Project aims to apply the knowledge acquired in SE/ComS 319, Construction of User Interfaces until Week 15. The project reflects the use of modern web development technologies commonly utilized by industry leaders.

Key Points:

- Although we set the minimum requirements of functionality and used technologies, the `Software` Final Project is a final product that solves a problematic of the preferences of each student.

- The Final Project is a Single-page website of perfect functionality and good aesthetic appearance. That is, the software must not fail, it must solve a fictitious or real situation or problem in a web environment and shows computer web pages with a professional appearance. Professional appearance can be interpreted from different points of view according to different students. For example, the appearance and design of a financial application does not necessarily reflect the seriousness and formality of a banking institution, but **it will be penalized if the appearance and design reflect very little effort and work**.

- It encompasses technologies covered in the course until Week 15.
  The addition of more advanced procedures and technologies used to solve technical problems won't be penalized. However, if the inclusion of more advanced procedures and technologies interferes with the review of the project at the time it is reviewed by the TA, the TA can decide whether to penalize the project or not.

Functionality

The Final Project must show 100% the functionality proposed in the Final Project `Proposal`.

Functional Requirements:

- Utilize a single-page web design with multiple modules as needed for different functionalities.

  The website is made up of different web views using the well-known concept of Single-web page. That is, the web site is developed in a main file using Javascript and multiple modules as necessary, which are imported to integrate multiple functionalities. Once committed to this requirement, is permitted to add more HTML files to solve more functionalities or problems.

- Ensure smooth navigation between views using mechanisms demonstrated in class.

  The user will navigate from one view to another without any difficulty or problem. The application will provide the appropriate navigation using the mechanisms or technologies shown in class, such as show/hide views or routing.

- Strive for a visually pleasing web appearance; creativity is encouraged, but an apparent lack of effort will be penalized.

  It is expected to have a good aesthetic appearance in the web pages. However, it is not specificized how good or bad is a good or bad design. We will give freedom to the creativity and skill of the students in this regard. However as mentioned in the Overview section, the excess of simplicity, the lack of effort to improve the appearance, will be penalized at the reviewer's discretion.

Requirements

Mandatory Technologies:

- HTML
- CSS
- JavaScript (including DOM)
- REACT
- NodeJS

- Express
- Unstructured (e.g., Mongo) or structured (e.g., MySQL) database.

Optional Technologies:

- Bootstrap
- Tailwind
- JSON

Key Software Elements :

- Use of hooks: useState and/or useEffect minimum
- Fetch or Axios to access external files and/or databases.
- Use of sync/await where necessary.
- The application must implement the **request methods:**
  - Post : **C**reate - Add one item at a time.
  - Get : **R**ead – Display all available items.
  - Put : **U**pdate – Modify at least one property of the item (e.g., price)
  - Delete : **D**elete – Remove an item given its id property.
- Use JSON data in external files for non-main data.
- But primary data should reside in the database.

App Views:

- The Application must contain at least 2 views for items or services.
  - Some students have asked how to meet this requirement if they are not developing an e-commerce application, and the response is that even constructing a web application of different problematics or services there are many ways to present the information using different views. Although, it is not a product catalog, the web application must show different views.
- An additional view is required to show student information, including course details:
  - Name of the course: SE/ComS319 Construction of User Interfaces, Spring 2024
  - Date
  - Complete name of students and ISU email
  - Complete name of Instructors and email :
    - ✓ Dr. Abraham N. Aldaco Gastelum
    - ✓ Dr. Ali Jannesari
  - This page does not require images; however, team member photos will be appreciated.
- If developed for a real company or institution, include a view from the company information.
  - The App must contain one view with the information of the Company if the project was developed for a real Company or Institution.

File and Folder Structure :

- Create a **"teamXX_final"** folder, to include all application components.
- Subdivide into “frontend” and “backend” folders for REACT and NodeJS development respectively.
  - Frontend
  - Backend
- In the folder **Frontend:**

All REACT development must be inside this folder.
All REACT and front-end configuration files must be inside this folder.

- In the folder **Backend:**
  All NODEJS + Express development must be inside this folder.
  All Database files must be inside this folder.
  All image files, text files, JSON files must be inside this folder.
  All NodeJS, Express, Database configuration files must be inside this folder.

3.- Documentation :

Additionally, to the software, the student must hand a Document of the system developed in Microsoft Word and exported to PDF.

The final website is made up of different web views using the well-known concept of Single-web page. That implies the system running will show different functionalities and interfaces to solve a problem.
On the other side, to develop the application was required to install software and construct directories and files.

The documentation will include the system architecture from the point of view of the physical structure of files, and directories. For example, the documentation will show the list of files and directories. If it is used a file containing data, the documentation will show the content of the data file; one portion of the data file will help to illustrate the information in the file.

The logical relationship among files will show the logical workflow of information/data flow. To illustrate the client-server relationship one or several diagrams of the communications among files can be used.

The documentation must include the next parts:

- **First page:**
  - Code of the course, name of course
  - Date
  - Students name, email address
  - Instructors name
  - “Final Project Documentation”
- **Second page:**
  - Index of the document
- **Sections:**
  - Project description
  - Software functionality diagram
  - **Files and directory architecture (physical and logical):**
    - Description and diagram of the files and directory architecture.
      - Diagram of the modules or blocks that represent user, server, intermediate files, CSV, JSON, database, Web pages, etc.
    - Description and diagram of client – server architecture OF YOUR PROJECT.
    - Description and diagram of the logical architecture.
      - Graphical representation of code/modules can be draw in tools such as Powerpoint, Figma, Excalidraw, Whimsical, etc
    - Database and API diagrams:
      - Diagram to describe the functioning of the API.

    - Explanation of each web view:

        - Screenshot of the main Web page and description of each field that show relevant information to the user and each field that one user can interact with.
        - Screenshot of views after interacting with the Interface using buttons, input, etc.

- Installation manual (Software and configurations):
    - Include a section of installation and configuration of software such as: NodeJS, React, Express, Bootstrap, Mongo, etc.
    - The manual must contain special configurations such as the important functions used to access external files, middleware used to access databases, etc.
- Copy of the code:
    - Copy of Javascript code
    - Copy of JSON files
    - Copy of HTML files

If your project is developed to a client and if for reasons of confidentiality it is not possible to write details of the data and of the system in general, for example: the cases in which the application was developed for a company, then an email from the staff of the company / institution requesting that this data not be published must be added to those documents.

4.- Video

The video is one essential part of the Final Project. A good video will show briefly, in 1.5 to 3 minutes, the physical architecture of the files and directories and the logical flow of data, the client-server relationship and the user interfaces.

Please use a video recorder that allows you to record the screen of your computer and the audio of your voice.

Do not use the cellular phone to create the video.

In the past, we have reviewed videos created using cellular phone and we have determined that are videos of poor quality, don't catch the screen completely and it adds background noise.

The format recommended for the video is `mp4`. Using a different format may cause the TA cannot play the video.

The video must include the student's voice providing insights.

Essentially, the video must show the most brief and quick possible, the files and folders structure, and the application in execution.

- When the application is executing, it must show the different views the application offers to the user. When entering data actioning buttons or interacting with interfaces,
- The video must show the outputs in web views and the possible modifications of data in the database.

The first recorded video may not be fully liked by the student and may not reflect the complete architecture and full operation of the application. Therefore, multiple recording attempts may be necessary for optimal presentation.

The instructor and the TAs will use the video to review the functional part of the Software, the completeness of the technological and procedural requirements and the integration of the minimum technologies that it must contain the software application.

5.- Demo :

During the first days of Week15, a file will be shared to register the name of the students according to a date and time the students will schedule a live demo to present their project to TA or Instructors.

In the final show demo presentation, each team member must exhibit a thorough understanding of the application's physical and logical aspects, code functionality, and technology usage.

What to Showcase:

- Show the backend and frontend code.
- Show the database and data.
- Demonstrate any user interactions in the frontend or functionalities you implemented.
- Demonstrate the access methods of the API: Get, Post, Put and Delete
- Discuss the technical aspects and challenges overcome during development.
- The TA or Professor can make questions about the files and directories architecture, or questions regarding the programming and code, and its functionality. Each student on the team must show complete knowledge of the application.

- The student must be able to answer questions related to software required, installation and configuration, knowledge of technology comprised on the Web site and functionalities of particular parts of the code. Even though the general work could have been divided for the development, both students must know 100% of the code, the technology used and its operation just as if they had done it individually.

During the demo session the TA and Instructor may review the Final Project Software, Documentation and Video. Therefore, the grade of these final assignments can be determined and made known to the students.

The show demo presentation should last no more than 30 minute per Team.

6.- Submission

Software

- Include all files in `teamXX_final.zip` (except MODULES folders).
  - Separate in a different folder the files pertaining to the Front-end and in another folder the files pertaining to the Back-end. Then make the Zip:
    - Frontend

- Backend

*Where XX in `teamXX_final` is your team number. That will allow the TA and instructor to review and assign the same grade to the team members.

*It is required submit ONLY ONE compressed file (.zip) per Team. Team members are grouped in Canvas.

Documentation

- Submit a PDF file containing all required documentation.
- File name `teamXX_final_documentation.pdf`

Video

- Submit a video file in mp4 format.
- File name `teamXX_final_video.mp4`

7.- The Rubric for grading this assignment is as follows:

This Assignment has a total value of 100 points.

There is no late submission deduction this time. There are not Extensions.

- Software ? _______ / 100

  - Is it developed using REACT ? _______ / 20

  - Is it developed using Nodejs and Express ? _______ / 20

  - Is it implementing CRUM Post, Get, Put, Delete ? ______ / 20

  - Is it implementing Hooks, Fetch/Axios ? _____/ 10

  - Is it developed using Databases ? _______ / 10

  - Is it a Single-web page application ? _____/ 10

  - Is it aesthetically appropriately developed ? _____/ 10

- Documentation ? _______ / 100

  - Has it First, Second and Index pages ? ____ / 5

  - Has it a clear description and diagram of the physical architecture ? ____ / 35

  - Has it a clear description and diagram of the logical flow of data ? ____ / 40

  - Has it screenshot of different views and explanations ? ____ / 10

  - Has it a software installation manual ? ____ 5

  - Has it copy of code, files ? ____ / 5

- Video ? _______ / 100

    - Is it clear, fluid, and easy to understand and was it recorded with a computer application ? ____ / 20
    - Does it show the physical architecture of files and folders ? ____ / 40
    - Does it show logical functioning of the application, showing views and user interactions ? ____ / 40

- Demo presentation ? _____ / 100
    - Is the student’s explanation direct and clear? ____ / 25
    - Does the explanation include the physical architecture of the system ? ____ / 35
    - Does the explanation include the logical data flow ? ____ / 40