**Final Project**

**Full Stack Development Bootcamp**

*Prepared by Jana Maher Kerbaj*

**Overall Grade: /100 Grade Distribution:**

**-30% Proposal**

**-70% Developing the Application/Website**

**Gr**

**Objective**

The main objective of full stack development projects is to expose students to real-life situations that software engineers have to deal with, developing a project spanning the different phases of software processes, ranging over: requirements engineering, design, development, testing, and deployment.

**General Procedure**

**Submission 1:**

Each student should orally present a proposal following the deadlines indicated below.

Contents of the proposal should include:

- A project title

- The project’s main objectives

- The project’s real (or potential) clients

- A short description of the project: What the project is about, why you want to do it (try to sell the project, providing convincing arguments: hot topic, relevant business needs, prospective applications, etc.), and how you plan to do it (using which SE process, programming environment, platform, preliminary plan of work, etc.). It is required to discuss the project topic with the instructor before presenting the proposal, to make sure it’s on the right track.

**Submission 2:**

*Duration of the Presentation:* 15 minutes for each student to pitch and present the system they developed.

**Suggested Project Topics**

1. **To-do List Project**

A very basic and simple project with a daily life use case is a to-do list app. A to-do list is a very popular full-stack project for beginners. You can build a to-do list, just like Microsoft’s to-do list application. It will have the feature to add an item to the list and if the task is completed then the functionality to remove it from the list should also be provided.

***The knowledge that you will gain from this project:***

This project will help you enhance your front-end development skills as well as storing the items in the database and deleting them from the database will help you learn the CRUD operations on the database in a great manner. You can also implement the user login and authentication for using the to-do list app. The link for a sample to-do list website’s source code is below. Feel free to take inspiration from it and use your own skills and creativity to develop an amazing to-do list.

1. **Blog Website/App**

A blog website is one where users can put up their thoughts about anything and everything. You can make it as creative as you want to. There are thousands of blog websites out there. You can take inspiration from any of these blog websites and create your own. Here, people have to log in using their credentials and proper authentication will also be done. After that, the user can put up their blogs for the public and they should be stored in the database so that it remains there. Also, the functionality of editing a blog after posting it and deleting it from the post would be a cherry on the cake.

***The knowledge that you will gain from this project:***

The front-end of a blog website should be really impressive for the bloggers to be attracted to your website. So, a lot of hard work and learning will go into the front end. Also, all the blogs will be stored in the database and if the blogger edits it, changes should be reflected in the database too. So, handling the database and making an attractive front-end are the main learnings that one can get from this project. The GitHub Link for a sample Blog Website with its source code is given below. You can take inspiration from it and try to make your own attractive and amazing blog website.

1. **Chat Application/Website**

We all chat with our friends and family daily. Chatting/messaging apps have become a part of our daily lives. Whether it is some important document regarding official work or a simple and delightful joke, everything is shared on the chatting apps. So, why not build one of your own? Chatting apps are also very popular full-stack projects. These applications (on a basic level) serve the purpose of group chats and chats between individual users.

***The knowledge that you will gain from this project:***

There is a lot to learn from this project. Firstly, the front-end should be decent looking as well as one very important thing to keep in mind while making the front-end for a chat application is the user experience. The user experience should be made as convenient as it can be. It should be easy for the user to send a message to a friend or a group and not a complicated procedure. Again, login and authentication are required for anyone to use this application. Also, the database should be managed well to get the latest as well as very old chat data. One very important thing is real-time communication between the users. This can be achieved using technologies like Socket.io. So, overall, there is a lot to learn from this project. The GitHub Link for a sample Chat Application/Website with its source code is given below. You can take inspiration from it and try to make your own attractive and amazing chat website/application and invite your friends to chat with you on your own platform.

1. **Portfolio Website**

Portfolio websites are a very popular concept in today’s world. The developers build some awesome portfolio websites to showcase their skills and impress the clients. Students or even professionals learning web development should make a portfolio website of their own as it is a great practice, especially front-end, and this is something where there are no limits. You can be as creative as you want to be. Links to some of the most amazing portfolio websites are given below:

***The knowledge that you will gain from this project:***

You will gain a lot of knowledge and experience in front-end web development by making this project. Also, every portfolio website has a “Contact-Me” section in which the user visiting your website will fill out the form to contact you. Here, the knowledge of the backend is required and the details of the users will be stored in a database. Overall, this project shows your skills as a developer and also you get great practice. The GitHub Link for a sample Portfolio Website with its source code is given below. You can take inspiration from it and try to make your own attractive and amazing portfolio website and showcase your skills.

1. **Food Delivery Application/Website**

Who doesn’t like food? Everyone wants to enjoy a delicious meal after the end of a hard working day. So, there are various websites and apps like Uber-eats and Swiggy, and Zomato in India, who have become popular and grown their businesses around this idea. So, a food application or website is another interesting full-stack project that is very popular as well as a not-so-old idea.

***The knowledge that you will gain from this project:***

You can learn a lot about developing a large-scale full-stack website or app by making this project. It involves a lot of backend knowledge and practice in making a major project like this. For instance, you require a backend for registering a restaurant under your services or ordering the food or changing the delivery address, and so on. Knowledge of the front-end is also a must and is going to help you a lot in making an interactive website/app. Managing a large-scale database is something that you can get familiar with by making this project. The GitHub Link for a sample Food Delivery Application with its source code is given below. You can take inspiration from it and try to make your own food delivery application with some delicious looking front-end and a butter-like smooth backend.

1. **E-commerce Website/App**

An e-commerce website/app is a very famous project and you will find it in almost every full-stack developer’s portfolio. This project is a very vast project and involves a lot of front-end, backend as well as database knowledge. It is considered a major project and it takes a lot of hard work and time to make it. You can either make an e-commerce site/app of your own or try to clone an existing large-scale app/website like Amazon or Flipkart, etc.

***The knowledge that you will gain from this project:***

This project takes up a lot of time and energy. You will learn a lot about how the backend of this e-commerce website works and a lot about user security, authentication, etc. Managing a huge database with different categories of products and different catalogs will also be a challenge. Overall, a lot to learn from this project. The GitHub Link for a sample E-commerce Website with its source code is given below. You can take inspiration from it and try to make your own ecommerce website.

1. **Video Conferencing App/Website**

The video conferencing apps and websites are also very good full-stack projects to try your hands upon. After the global pandemic struck us, video conferencing apps like Zoom, Google-Meet, Microsoft Teams, etc have been a major part of our lives. The features of a video conferencing app can be audio-video chatting, textual chatting, recording the video and saving for later use, saving the chats during a call too, a whiteboard to draw and explain, and so on. There is a lot to implement and learn in this project. You can implement your own ideas to make it more creative. For instance, apart from the features that are there in any other video conferencing app/website, you can add the feature of a code-editor which can be used by the companies during the online interviews so that the candidate can write the code by himself/herself.

***The knowledge that you will gain from this project:***

One important thing to note here is that this project is not going to be as easy as making a simple textual chat application. It involves a lot of extra work. The feature of implementing an audio-video call and managing it correctly is itself a big task. Apart from that, the video and audio conferencing, drawing on the whiteboard such that everyone can see your point/opinion, textual chat, emotes, etc should all be real-time (again you may use Socket.io for it). You will also have to think about how you can add the feature of recording the video call and saving the chat. So, there is a lot to learn from this project. The GitHub Link for a sample Video Conferencing Application with its source code is given below. You can take inspiration from it and try to make your ZOOM or your own GOOGLE MEET Application.

1. **Social Media App/Website**

Social media is all that we are surrounded by in today’s world. There are a lot of social media apps and websites that we use in our daily lives like Instagram, Facebook, Snapchat, etc. A very good full-stack project can be developing your own social media app. A social media app/website is not like a chat application. Here, people can post an image/video/text, and others can like/dislike, comment, and share them. It is like forming a network of people. This project is a very good project for the practice of both front-end and back-end (along with database).

***The knowledge that you will gain from this project:***

You will have a good practice of front-end development because a social media app/website has to look attractive for the users to use it regularly. Also, the backend needs to be pretty good. You can also use encryption techniques to encrypt the messages of different people. Overall, lot of learning from this project too. The GitHub Link for a sample Social Media Application with its source code is given below. You can take inspiration from it and try to make your own social media application and become the Mark Zuckerberg of your own world.

1. **Content Management Tool/System for Blogs**

WordPress is a good example of the Content Management System. The user can drag and drop items available to create a web page. Such a project is not very common and can count for your uniqueness as a full-stack developer. Users in a Content Management System (CMS) must be able to create web pages of their own by drag and drop method. Such features should be added so that the user should be able to add textual content and media content into placeholders attached to the web page. So, the users should be able to design the websites and publish blog posts. This can happen using a text editor that accepts the input text from the user and converts it into HTML and pushes it into the database.

***The knowledge that you will gain from this project:***

The knowledge of the front-end will be tested very rigorously in this project. Also, back-end knowledge will be required for uploading and saving the blog posts. Overall, this will be a unique project as well as you can learn a lot from it.

1. **Project Management Tool**

The toughest full-stack project and one with a lot of features is the Project Management tool. Examples are Kissflow Project, Trello, Asana, etc. This project should have the framework of a basic social media site with users being able to communicate with each other. Users should have the facility to create projects together as well as create tasks within the projects. The users should also have the functionality to assign tasks to one another and also comment on it like social media.

***The knowledge that you will gain from this project:***

You will gain a lot of knowledge of front-end, back-end, database, and whatnot. You will even learn how the projects are assigned to a team, how the work is done in a team and how professional and industry scale projects are built. There is a lot to learn in this project and it is going to be difficult to develop this project independently. It is recommended to develop this project in a group. The GitHub Link for a sample Project Management Tool Project with its source code is given below. You can take inspiration from it and try to make your Project Management Tool. A good project management tool is always in demand in the software industry.

1. **Calculator**

A calculator is a great chance to develop your layout skills. Calculators also require some input/output knowledge, as you need to capture data from the user's peripherals. Your calculator app should:

Depict the visual element of a calculator cleanly on any platform

React to the user's input, both through the mouse and keyboard

Save the user's history, even if it just does so in a text file

1. **Weather App**

What's the weather like outside? Don't look out the window. Program an app instead. Your weather app should:

Pull weather information from a database

Change its appearance based on the weather

Use an attractive stylesheet to deliver hourly and weekly forecasts

As an additional challenge, connect your app to a weather API to pull real-time weather information based on the user's location. There are free weather APIs available that you can easily connect to through JavaScript.

1. **Quiz App**

This is probably one of the most useful full-stack coding project ideas for portfolios; you can even use it to quiz yourself on full-stack development. Your quiz app should:

Pull quiz questions and answers from a database

Display questions to the user and prompt them for an answer

Track and retain the user's score, eventually letting them "win" or "lose”

1. **Interactive Resume**

Why not give potential employers a treat? Build an interactive resume that doesn't just display your skills, but also prompts the user to inquire for more information. Your interactive resume should:

Display your resume in an attractive, mobile-friendly format.

Prompt users to send you their information (if they want to learn more).

Show off dynamic, interactive CSS assets, such as parallax.

1. **Music Streaming App**

A music app is one of the most common full-stack app ideas. Everyone loves music. But what can you do to make your music streaming app better? At its most basic, your app should:

Pull music from a database of songs

Provide a music streaming interface

Let users either select songs on a playlist or search for songs

To challenge yourself, build a suggestion engine. Track what users play and suggest similar songs based on their preferences.

1. **Job Board**

Maybe you don't want to build full-stack portfolio projects. Maybe you want to help other people get jobs. A job board can be a fun project, although you need to keep it updated. It should:

Pull job information from a database

Display job information based on search criteria and filters

Let users save certain jobs or even respond to them

If you want to make this job board autonomous, consider programming it to scrape information from other sites.

1. **A Real-Time Game App**

Do you have a burning idea for a game that you've passionately pursued full-stack development to achieve? The time is now. To qualify as a game, your project should:

Support user accounts and user data through a database

Provide some interactivity or gameplay elements

Have clear goals or provide some state of "winning" or "losing"

Provide a profit model, either through subscription or in-app purchases

Operate both as a desktop application and a mobile application

1. **Workout Tracker App**

Tracking workouts or exercises puts an interesting spin on a regular scheduling app. You can do this with any type of recurring hobby. Develop an app that will:

Support user accounts and customization with authentication

Schedule workouts/hobbies and reschedule missed sessions

Let users determine the frequency and preferred scheduling of their workouts

Connect users through support groups and competitive teams

Intelligently send users reminders and alerts

1. **Image Sharing App**

You've seen Instagram and Imgur. It's time to make your own image-sharing app. Image sharing isn't just popular; there are a lot of potential niches to fill.

An image sharing app:

Let users upload, store, and manage images

Provide filters so users can update and edit their images

Compress images intelligently for better performance

Automatically prune unused images of abandoned images

Let’s users pay for more space on or other premium features

1. **Collaborative Editing App**

This full-stack project tests whether you're ready to create a real-time application. A collaborative editing app will need to:

Support multiple users in a single room editing a file

Alert users if they are trying to edit something someone else has edited

Combine edits intelligently or lock files when they are being edited

Provide easy roll-backs to prior versions of a file

Authenticate users and support multiple permission types (such as viewer and editor)