Shiksha - Frontend Restructuring

Siddhant Tohan

Netaji Subhas University of Technology

Synopsis

The main aims of the project are as follows:

1. Split code into components and Micro Frontends using module federation.
2. Shift the codebase from Javascript to Typescript.
3. Follow a Test Driven Development(TDD) approach using frameworks like jest and mocha.

This proposal describes in great detail how the given objectives can be completed by following the best practices and keeping in mind the future feasibility of the app. We will be using a variety of modern frameworks like React, Framer Motion, Redux, and much more to make a very lightweight yet efficient app that can provide a great experience to students and teachers alike.

Relevant Skills and Motivation

Skills

1. My languages of choice include:
    - HTML5
    - SASS
    - Javascript
    - Typescript

The given languages are the most widely used languages in web development. Javascript especially is currently the most popular language used for both frontend and backend development. Typescript is a compiled variant of Javascript that helps to write more robust code. As the project involves migrating from javascript to typescript, having good knowledge of both is absolutely necessary.

SASS is a preprocessor of CSS that makes styling much easier and modular.

Motivation

My main motivation to apply for this project is to improve the state of our national websites. These days whenever anyone hears the word “Government website”, their first impression is an outdated looking and buggy website that looks like it's from the 2000’s. These days many changes are indeed being made to our websites and public opinion is slowly improving, and that’s why I wish to join in on this movement and help to make more modern looking and highly functional websites.

This also gives me an opportunity to contribute to a product that will be used by many teachers and students of our country on a daily basis.

Project Details

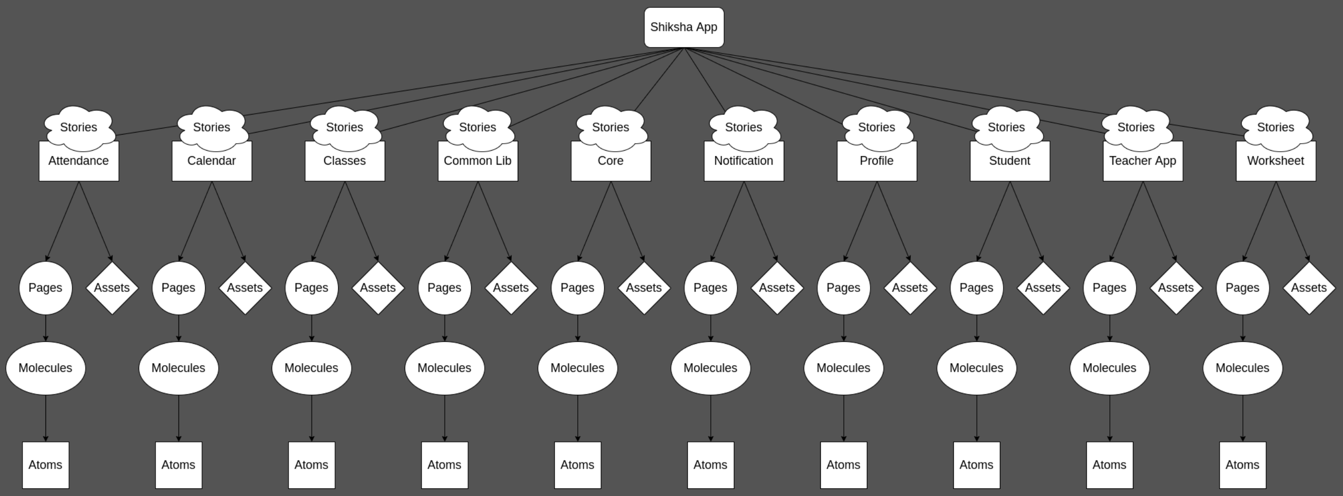
I. Overview

a) Working:

The project will work in the following manner:

1. The app as a whole is divided into many different modules. It follows the micro frontend development approach, where each standalone module can also function independently and can be used in other websites also as per requirements.
2. Each module consists of a number of pages, which are navigable parts of the module. For eg, in Worksheet Module, the question banks and view worksheet can be treated as two different pages.
   - ❖ Each module also contains all the assets(like images or svgs) which can be imported and used in its pages.
   - ❖ In each module, we also use stories to make things the code more robust
3. Each page consists of a number of molecules, which are small, configurable parts that can perform some function and are also capable of data mutation.
4. Each molecule consists of multiple atoms, each atom is like a dumb component that performs very primitive functions with little to no support of data mutation.

Here is the overview of the app as an whole:


Shiksha App
Stories
Attendance
Stories
Calendar
Stories
Classes
Stories
Common Lib
Stories
Core
Stories
Notification
Stories
Profile
Stories
Student
Stories
Teacher App
Stories
Worksheet
Pages
Assets
Molecules
Atoms

b) Problems:

1. The most common problem when shifting from javascript and typescript is to insert the proper data types at their required places, whilst making sure these types that are defined are as reusable as possible.
2. Adding animations brings a very huge risk: backward compatibility and performance.
3. We also need to have a good amount of unit tests that test the various aspects of our code base.

c) Solutions:

1. By using the recommended guide here, it is quite easy to ensure the code doesn’t break when shifting from javascript to typescript.
2. It is always recommended to use CSS animations for good performance. Libraries like framer motion and React Transition Group use css animations by default, which is a big help. For backward compatibility, we may use polyfills for unsupported code and we can also configure our babel transpiler to get the desired results.
3. By using tools like jest or mocha, it becomes very easy to test the various aspects of our code in an automated manner.

II. Implementation

For this project, we will be using the following tech stack:

1. HTML
2. CSS
3. SASS
4. Typescript(with React)
5. React Stories
6. Framer Motion / React Transition Group(for animations)
7. React Router
8. Jest / Mocha(for unit testing)
9. Cypress(optional, may be omitted if E2E is not required)
10. Lerna and Yarn(for building and installing dependencies)

1. Milestone 1

First step would be deciding on the distribution of workload to the contributors. We would also decide what **additional frameworks to use in addition to the current ones.** Once these are decided, we will check their compatibility with the current codebase and adjust accordingly.

Expected Duration: 2-3 days

2. Milestone 2

In this phase, **we will be focusing on migration and refactoring.** We will also be writing unit and integration tests for each module. This will help to produce robust code.

For the migration part, we will use the TypeStat tool to automate conversion from JS to TS. However, it has certain limitations and will not complete the migration in one go. We may need to fix some things ourselves.

The full guide for TypeStat tool can be found in their Github repo.

Expected Duration: 18-20 days

3. Milestone 3

For the 3rd milestone, we will be working on **adding animations(including reveal animations) and also the styling.**

For animation,the decision between framer motion and react transition group largely depends on the contributors, Framer motion has an easier learning curve but React transition group is the official way of animating things in React.

For styling, we will use the SASS preprocessor to enhance the styling.

Expected Duration: 15-20 days

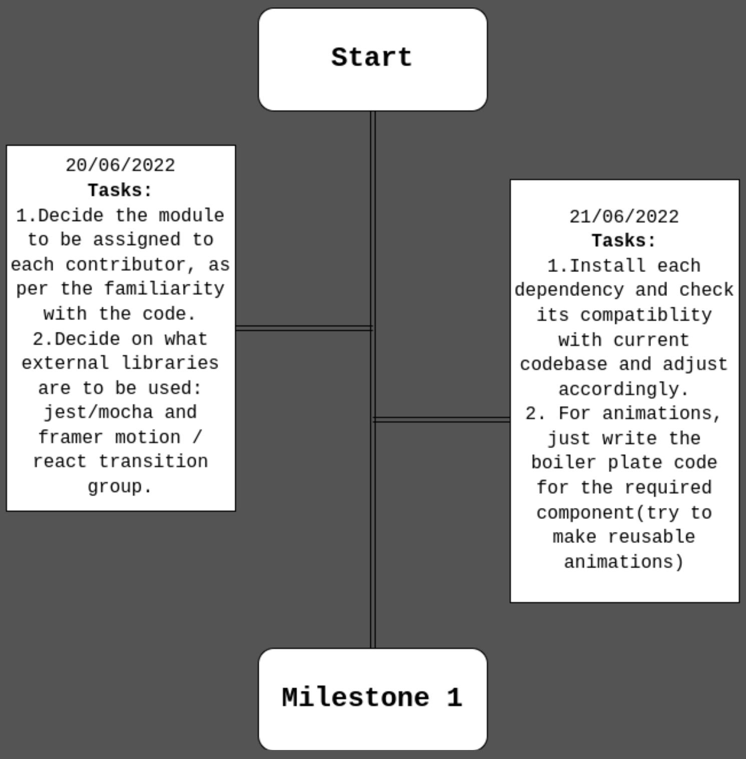
Timeline

Start Date: 20th June, 2022

End Date: 31st July, 2022

PTB for the flowchart describing in great detail the exact workflow while making this project.

Phase I



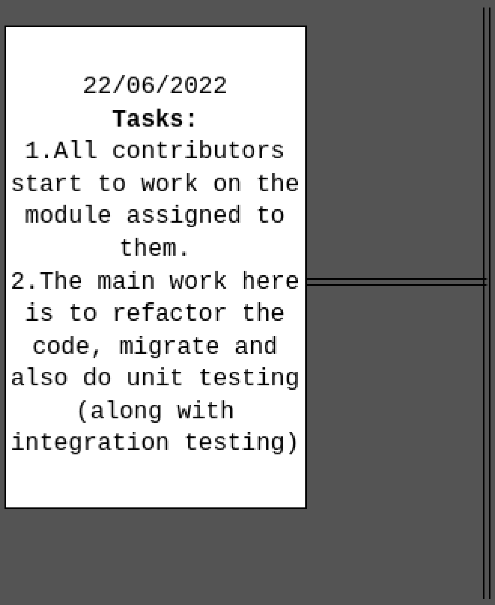
Start

20/06/2022
Tasks:
1.Decide the module to be assigned to each contributor, as per the familiarity with the code.
2.Decide on what external libraries are to be used: jest/mocha and framer motion / react transition group.

21/06/2022
Tasks:
1.Install each dependency and check its compatiblity with current codebase and adjust accordingly.
2. For animations, just write the boiler plate code for the required component(try to make reusable animations)

Milestone 1

Phase II

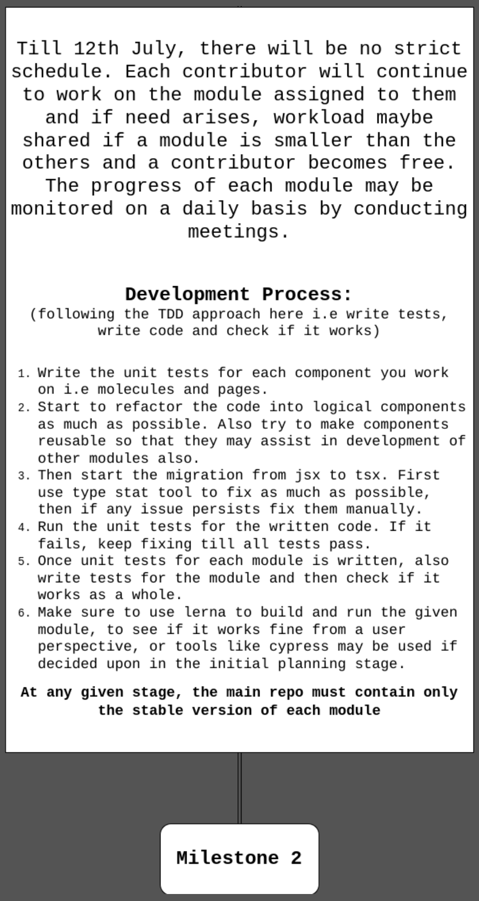


22/06/2022

Tasks:

1.All contributors start to work on the module assigned to them.

2.The main work here is to refactor the code, migrate and also do unit testing (along with integration testing)



Till 12th July, there will be no strict schedule. Each contributor will continue to work on the module assigned to them and if need arises, workload maybe shared if a module is smaller than the others and a contributor becomes free. The progress of each module may be monitored on a daily basis by conducting meetings.

Development Process:

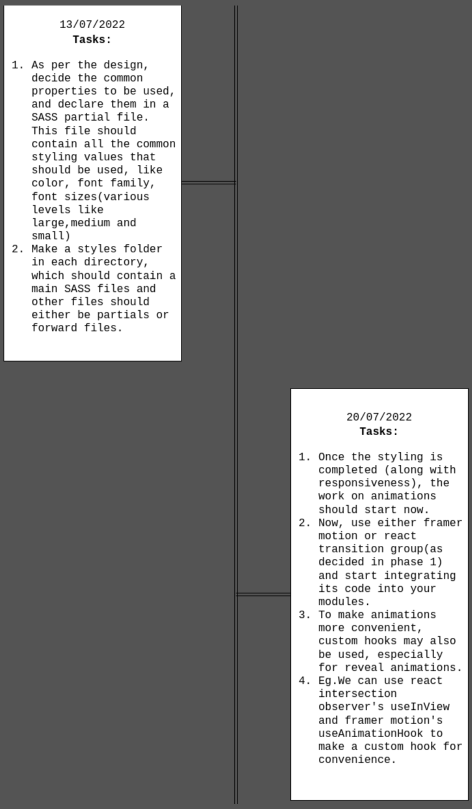
(following the TDD approach here i.e write tests, write code and check if it works)

1. Write the unit tests for each component you work on i.e molecules and pages.
2. Start to refactor the code into logical components as much as possible. Also try to make components reusable so that they may assist in development of other modules also.
3. Then start the migration from jsx to tsx. First use type stat tool to fix as much as possible, then if any issue persists fix them manually.
4. Run the unit tests for the written code. If it fails, keep fixing till all tests pass.
5. Once unit tests for each module is written, also write tests for the module and then check if it works as a whole.
6. Make sure to use lerna to build and run the given module, to see if it works fine from a user perspective, or tools like cypress may be used if decided upon in the initial planning stage.

At any given stage, the main repo must contain only the stable version of each module

Milestone 2

Phase III



13/07/2022

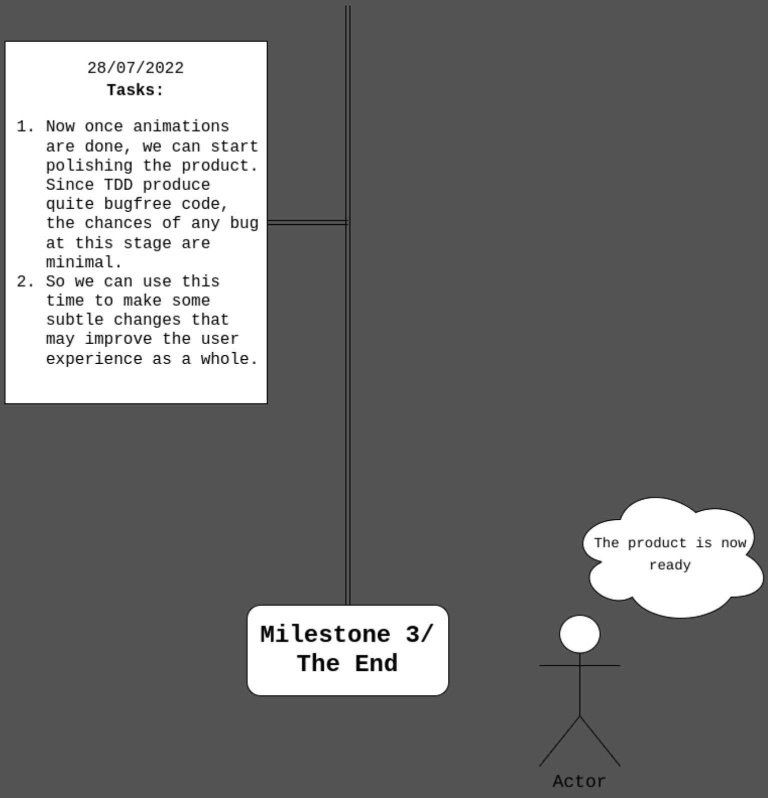
Tasks:

1. As per the design, decide the common properties to be used, and declare them in a SASS partial file. This file should contain all the common styling values that should be used, like color, font family, font sizes(various levels like large,medium and small)
2. Make a styles folder in each directory, which should contain a main SASS files and other files should either be partials or forward files.

20/07/2022

Tasks:

1. Once the styling is completed (along with responsiveness), the work on animations should start now.
2. Now, use either framer motion or react transition group(as decided in phase 1) and start integrating its code into your modules.
3. To make animations more convenient, custom hooks may also be used, especially for reveal animations.
4. Eg.We can use react intersection observer's useInView and framer motion's useAnimationHook to make a custom hook for convenience.



28/07/2022

Tasks:

1. Now once animations are done, we can start polishing the product. Since TDD produce quite bugfree code, the chances of any bug at this stage are minimal.
2. So we can use this time to make some subtle changes that may improve the user experience as a whole.

Milestone 3/ The End

Future Development

After this program concludes, I will continue to work on full stack apps with many startups, like I am doing right now by working with a Web3 based startup. I have also been getting interested in Open Source lately, so I will also try to contribute to many repositories with issues that I believe I can solve with my skill level.

If allowed, I would be willing to work on this app even after the program concludes, and continue to make improvements by being in touch with mentors.

Availability

1. Currently my vacations are ongoing till the end of this program(July 31st). So I will be available for most of the day.
2. Currently I am working with a Web3 based startup, but the timings are flexible and therefore it would not interfere with this project. I have good faith in my ability to balance load appropriately.
3. I will be able to devote 4-5 hours every day, that is 28-35 hours per work at the least. If there is a requirement for more, I will be willing to work more without any questions.

About Me

My name is Siddhant Tohan, currently I’m 20 years old. I am known to be a calm and collected person by my colleagues. I am always willing to lend a hand to any of my colleagues if they need help. I like to listen to music, coding and playing games.

I am currently enrolled in Netaji Subhas University of Technology. I am pursuing a degree in Computer Science and Engineering with specialization in Mathematics and Computing. I will graduate in 2024.

I achieved 96.6% in CBSE 2020 Class XII exams. I also achieved 94.8% in CBSE 2018 Class X Exams. I pursued Non-Medical Science in my class XII.

I am currently working as a full stack intern at BlokMiners, mainly with React for now. I also have achieved grade 4 out of 5 in Full Stack Open, a course by University of Helsinki for full stack development.

I only recently got into open source and have made some PRs on public repositories, including the Shiksha Frontend Repository. I am very familiar with git VCS and have more than 700 commits(you may check my profile here).

I believe that I am an ideal candidate for this project as I have a lot of experience in developing apps. I am very confident in my ability to write clean and robust code in a short period of time. I also pay attention to minor details that many people generally overlook.

Thanking Note

I would like to thank the mentors who have stepped up to make open source culture popular in India. I would have never thought I would see the government itself sponsoring an open source program like this. I would be willing to stay connected with everyone to learn more in the future.

Thanks to everyone who made this program possible.