
Institution Details

Province	Sindh	City	Karachi
Institution	The National University of Computer and Emerging Sciences (FAST-NU)	Campus	Karachi
Department	Computer Science	Degree Level	BS
Degree Program	Computer Science	Telephone	
Fax			

Supervisor Details

Name	Muhammad Rafi	Gender	Male
Mobile	+92 300 2326840	Office No	CS Department, 111-128-128 (Ext 222)
Email	muhammad.rafi@nu.edu.pk	Designation	Assistant Professor
Qualification	PhD		

Co-Supervisor Details

Name		Gender	
Mobile		Office No	
Email		Designation	
Qualification			

Head of Department Details

Name	Dr. Zulfiqar Memon	Mobile No.	
Email	zulfiqar.memon@nu.edu.pk	Gender	Male

Project Details

Project Title	CodeClassy		
Group Details Project Area of Specialization	Member 1 Name: Saif UI Islam Member 1 Roll#: 18K 0307 Member 2 Name: Muhammad Hassan Zahid Member 2 Roll#: 18K 0208 Member 3 Name: Tashik Moin Sheikh Member 3 Roll#: 18K 0142 Machine Learning, Software Engineering, Distributed Computing, Web Application		
	Project Start Date September 2021	Project End Date July 2022	
Project Summary (< 2500 characters)	CodeClassy is a product aiming to provide a flexible, heavily featured, organized, easy-to-use, and integrated system to counter many challenges found in EdTech . With the COVID 19 pandemic serving as a baseline motivation, the lack of a general-purpose and the need for interoperated systems was realized to solve globally recognized problems for EdTech . While interests in modernizing education through technology had been soaring before the pandemic, the crisis presented a comfortable discussion and opportunity to develop around the lack of fully-featured, well-integrated, and dedicated systems to ideally work with the desires of educational institutions, matched with a/an (a) Ease Of Use , (b) Unified Platform , (c) Feedback System , (d) Collaborative Model designed primarily for engineering students to deal with online vs hybrid challenges. The result shall be an MVP built as a platform orchestrating a submission, review, and feedback through base functionalities.		
Project Objectives (< than 2500 characters)	CodeClassy has been inspired thanks to the experiences of the students, as well as from Professor Muhammad Rafi. The central core value is to provide a solution that deals with deeply observed problems, disorganization related to classrooms,		

monitoring student progress, providing feedback/questions, reviewing student solutions, keeping up to date, overall making the learning process feel tiresome rather than an enjoyable process.

The idea is for a concrete platform that would be able to provide “feature” extensions to traditional classrooms, making the best of remote learning and virtual classrooms, while focusing heavily on collaboration, incorporating many features together that are right now diversely spread in EdTech (think GCR, HackerRank, ClassMarker).

For us, integration meant features focused on,

1. Easy to use
2. Tackles popular obstacles and issues
3. Content organizing facilities
4. Automation where possible
5. Cost & time effectiveness

For us, it is vital to focus on developing, combining the above priorities into a single web application-based software project.

The aim is to release a valuable product, not only to us as an FYP, but as something with the possibility to be used globally by educational institutions, teachers, and students. We want to present a user-oriented design that provides plenty of resources as features most classrooms consider essential to great learning, and more. Our idea wishes to remove frustration, demotivation for students pursuing an online education.

**Project
Implementation
Method
(< 2500
characters)**

The project implementation will be based on two core use cases identified mainly in the project. These are,

1. **Quizzes** - designing an interface that deals mainly around the “quiz” aspect. Contains has use cases such as (A) Question Creation, (B) Combining Results,

(C) Exporting Results, (D) Handling Multiple Question Types, (E) Tracking "Groups"

2. **Remote Code Execution** - this allows students to work on developing and getting outputs for coding assessments. Contains use cases such as (A) Programming In-Browser, (B) Code Submissions, (C) Being Assigned Coding Assessments, (D) Presenting Outputs From Code

The plan is to target the "Quizzes" assessment, moving onwards with developing an MVP around the core implementation and logic during FYP I. We then move towards handling the 2nd core requirement in FYP II.

As for the implementation, we will start by dissecting use cases into smaller modules to work and develop on, and then moving on from there to integrate smaller components into a larger module. We plan to do the same for each "bigger" requirement. Even so, we may want to apply the process of first extracting the requirements, then designing, then implementing, testing, evaluating, over and over until we finally get the right results as we expect.

Testing

This section refers to testing modules when they are submitted by a team member.

We may or may not follow a TDD approach due to time limitations, and a semester along with this project. "Software Engineering" best practices promote TDD as a guide for testing, but ultimately the decision comes down to how long or how easy it is to write and promote testing.

Evaluation

Evaluation is a manual process for when a main requirement or module of the project completes. Each of us will attempt to use and modify the module, seeing if it is flexible and workable for our needs or needs changing.

The process can vary from module to module and is tricky to generalize. An attempt can be repeating the same steps as the original developer. This idea makes sure it does not break on the addition and removal of code as the project keeps processing.

By taking our product through a sustainable software philosophy, we will cater to evaluate code quality, maintain clean architecture, simplify and provide ease for future modifications to be of minimal impact to other existing features, thus preventing and lowering chances of breaking changes.

**Benefits of the
Project
(< 2500
characters)**

There has been a steady trend of working towards modernizing classrooms, both on a lower secondary standard, as well as on a University and College level to promote better technology and information literacy. Back in 2015, **56%** of students worldwide reported the desire for instructors to enable them to use their devices in the classroom [1].

Even more so, in 2015, **79%** of potential learners in the United States were unaware of massive open online courses [2]. Since then, there has been a stronger push towards trying to make students more familiar. This is directly shown with the expected growth of the EdTech domain.

In 2016, the self-paced e-learning industry made up **\$46.67** billion of the global market share and, by 2022, that number is projected to exceed **\$243** billion, from [3].

Despite this, there are still many pressing concerns. The COVID pandemic, initially serving as a base motivation in the start, has revealed intriguing figures as to how better software is needed.

9 out of 10 teachers reported that they spent more time troubleshooting technology in response to the COVID-19 school closures in 2020 [3]. In a remote position, it is harder to see what students are struggling with and what they may need help against.

A study by the NCES revealed that **19% - 28%** of US teachers respectively report that they assign technology-based homework often or at least sometimes [4], indicating that teachers were unable to evaluate as they might have done otherwise, and were then unable to potentially give feedback over the progress.

Alarming, As of Summer 2020, **93%** of US teachers reported conducting at least some of their teaching online from home [3].

It is evident that the market and the need to develop more fully-featured EdTech software and applications are needed that target this niche to try and solve the technical problems amongst the Educational flora.

The above mentions reveal such problems in EdTech as,

1. **Technicality Overhead** - software being used for education wasn't designed for remote education in the first place, resulting in teachers facing technical difficulties
2. **More Than MOOCs** - students demand more than just MOOCs that help them with a certification, and would like to vest time getting to know other students more through collaborative rather than isolation-based learning
3. **Instructor Training** - there is a gap between the level of training that instructors should have, and what they have. This can come from a lack of funding for more enterprise, paid and handled software in schools, universities, and colleges. There is no flexible solution helping to bridge the gap

In contrast to all these problems, CodeClassy, in its MVP stage, wishes to tackle the above problems as,

	<ol style="list-style-type: none">1. As discussed in previous sections, we aim to design an easy to use, flexible, and user friendly2. Collaborative learning, progress tracking, and staying organized with course material is a priority in the project. While certification from a MOOC is great, this feature can be implemented in CodeClassy further down the road if interest ensues and time allows3. Because we aim to make a web application accessible to everyone, only limited by an account, this will bring ease in adoption for instructors
Technical Details of Final Deliverable (< 2500 characters)	<p>This section is a bit vague. We have interpreted it as “Workflow”, “Project Management (GitHub)”, “Requirements”, “Features”, and “Technologies”.</p> <p>The features around CodeClassy revolve around leading assessments designed for either “<i>Formative</i>” or “<i>Summative</i>” assessments. A description of both follows as below,</p> <ol style="list-style-type: none">1. Formative Assessment - The purpose of formative assessment is to monitor student learning and provide ongoing feedback to staff and students. It is an assessment for learning2. Summative Assessment - The goal of Summative Assessment is to evaluate student learning at the end of an instructional unit by comparing it against some standard or benchmark <p>This leads to our current list of evaluated requirements,</p> <ol style="list-style-type: none">1. Quizzes2. Remote Code Execution <p>Quizzes</p>

As for the requirements, this includes “Question Bank”, “Question Markup Language”, “Tests”, “Test”, “Question”, “Answer”, “Group”, and comprises more than 15+ use cases, revolving around two main roles, “Teacher”, and “Student”.

Remote Code Execution

As for the requirements, this includes ...

Workflow

The planned workflow for the project will be based on a scrum-based approach, accordingly to which we will design and work around “sprints ” aimed at targeting and working on specific aspects of the project.

This method will allow us to easily plan what work needs to be done and what remains and will help the team to keep a self-check on the project’s progress so far.

Project Management (GitHub)

Our intended project management tool will be a combination of Notion along with GitHub. For each,

1. **Notion** - as it provides an easy way of documenting, writing down notes from meetings, having and serving as a place of discussion
2. **GitHub** - the team will most likely wish to use the following features as a project on GitHub,
 - **GitHub Actions & GitHub Marketplace** for automation through apps and GitHub actions that help in dealing with managing the project
 - **Kanban Board** from the “project” tab
 - **Wiki** if there’s anything worthy of discussion putting over there

- **Issues** to help with pointing out bugs, feature requests built tasks that can easily be assigned to team members
- **Pull Requests** to keep a track of new changes to the code base, ease of evaluation
- **GitHub Security** for setting up customized security policies, security advisors, Dependabot alerts, and code scanning alerts

Technologies

We're aiming to make something that is in production and can be used. We thus need technologies and a process to streamline the whole project to better cater to iterative development, feedback, automation, rapid/lightning-fast development, and enough infrastructure to handle basic workloads/requests from users and catering to software design principles of modularity, security, maintenance, and logging.

Thus, we split our technologies into different levels to work with the requirements,

1. Frontend: NextJS (TypeScript), Redux, Material UI (most likely) + NPM Libraries
2. Backend: NestJS (TypeScript) + NPM Libraries
3. Databases: MySQL + To be decided
4. DevOps: GitHub Actions, Docker, K8s, Docker Compose, Azure Services, Jenkins (+ Plugins), AKS, Docker Hub Registry, Ingress, Traefik
5. Testing: To be decided
6. Documentation: To be decided

**Final
Deliverable of
the Project**

As per the requirements for the semester, the deliverable of the FYP will include,

1. 30% prototype for FYP-I
2. 100% for FYP-II

FYP I

On the documentation aspect, FYP I will be submitting this "FYP Project Proposal " for the defense, and the "Software Requirement Specification". As for the features, FYP I will largely tackle "Quizzes requirement as the deliverable, and will cover the "Data Modelling " & "Entity Extraction", and features for Quizzes as discussed in "Methodology, Implementation". Overall, this phase includes working with the,

1. Requirement Analysis
2. Design Phase
3. Data Modelling
4. Architecture & Infrastructure Setup
5. Implementation Phase (WIP) & Boiler Plate Setup
6. Quiz Requirement

FYP II

FYP II will move towards the finalization of the FYP, as well as delivering towards the "Remote Code Execution" requirement.

Core Industry	EdTech
Other Industries	Education, Technology, Collaborative Platform, Community
Core Technology	Full Stack Web Developments, DevOps, Cloud, Computer Networking, Security, Software Architecture
Other Technologies	JavaScript, Docker, Kubernetes, Azure
Sustainable Development	Mentioned already under "Technical Details of Final Deliverable"
	This is undecidable (as enterprise this should be) according to "Sustainable Software Development" (SSD). The main aim of an FYP is to deliver as fast as possible for the

Goals

sake of developing a PoC (if needed) or an MVP (Minimal Viable Product). With short deadlines and feature commitments, the argument of dealing with preventive measures in software that has not yet hit user and production use, it seems unreasonable to plan completely according to SSD and “Sustainable Development Goals” (SDG).

However, it is indeed our intent to work with an approach that cuts down technical debt, makes it easier for us to add and remove features that we like, tweak configuration settings where we would need to, and swap old solutions with new ones if we ever come to that. For that, we need to practice and develop great software engineering practices and build upon ideas of clean code, architecture, and feasibility both in terms of time and resources.

This means for us the following,

1. Only building what is required to be build
2. Be proactive in solving problems, not reactive. It's much better to think through solutions rather than to build hacky ones that need patching frequently
3. Working smarter by automating tasks that reduces the load on the time we have
4. Reviewing code contributions to the project by each of the team members, to ensure context and clean code delivery
5. Following best practices where possible
6. Using formatters and linters to detect, and fix bugs, and present a standardized codebase

We intend to use the aforementioned tools as mentioned under “Technologies” in the above section to better cater to the project requirement needs that we will come across.

The health of our product is crucial for us because we stand at the desire to make it marketable and usable by people. We will treat security and implementations as serious concerns for the project, and ensure that rapid development stays constant over the next coming months.

References

- [1] E. Duffin, "E-learning and digital education - statistics & facts," Hamburg, Germany, Tech. Rep., 2020
- [2] "Global student awareness of massive open online courses (MOOC) as of April 2015," Hamburg, Germany, Tech.Rep., 2015
- [3] K. Bushweller, "How covid-19 is shaping tech use. what that means when schools reopen," Education Week -Special Report, June 2020
- [4] "Teachers' use of technology for school and homework assignments: 2018–19," IES, USA, Tech. Rep., 2018-1

Project Key Milestones

Elapsed time in (days or weeks or month or quarter) since the start of the project	Milestone	Deliverable
Month 1	Meetings and Project Exploration	NA
Month 2	Data Modelling and Wireframe Development	Design mocks and Database Schema
Month 3	Database setup, conversion of wireframes to user interfaces, documentation	SRS/SDS
Month 4	Start implementation of business use cases	30% of proposed system
Month 5	Adding more functionality for, 1. Quiz Management System 2. Remote Code Execution & Real Time Collaboration for writing saving, running, submitting code, while evaluating test cases	Working prototype/use cases
Month 6	Adding more functionality for, 1. Quiz Management System 2. Remote Code Execution & Real Time Collaboration for student grouping, assignation, result collection, and providing boilerplate	Working prototype/use cases
Month 7	Completion of business use cases	Working project deliverable
Month 8	Evaluation, Testing, Project Wrap Up	100% of proposed system

Project Equipment Details

Item(s) Name	Type	No. of Units	Per Unit Cost (in Rs)	Total (in Rs)
Workstation At University, Development Purposes	Machine	1	Already Available, NA	0
			Total in (Rs)	0

Close