**DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY**


**A Project Report**

**On**

**"Question Paper Generator System"**

**MASTER OF SCIENCE IN INFORMATION TECHNOLOGY**

**(Batch 2019-2021)**

*Submitted To:*                                      *Submitted By:*

*Dr. Shruti Bhilare*                                 *Jeni Biliyawala – 201912089*
*(Assistant Professor)*                              *Vasani Vibhuti - 201912110*

# 1. INTRODUCTION

## 1.1 PROBLEM STATEMENT

Every faculty in an institute need to make a question paper for conducting the exams at least once in a semester or at most thrice be it either objective or subjective or combined. Also, mostly every institute conducts the entrance test every year as well, which contains on an average 50 objective questions using different topics depending on the institute's requirements. This requires a lot of time for a faculty to select the questions from different topics of their respective subject keeping it balanced i.e. neither too hard nor too easy. Then all the questions are combined together to form a final question paper which may not be as trivial as it seems, eating up a lot of time and efforts.

## 1.2 PROPOSED SYSTEM

The question paper generator system is being developed to solve the problems prevailing in the manual system. This system allows eliminating and, in some cases, reduce the complexity caused during constructing such examinations. No formal technology/knowledge is required to operate the system.

## 1.3 PURPOSE

This software aims to simplify the interest of the institute and faculties who wants to conduct the exam under a quick notice or would like to automate this process. of making the question paper and who finds that drafting an unbiased paper requires a lot of time and effort.

On the other hand for the students it is a good way of getting the mock question paper to practice.

## 1.4 SCOPE

Scope is to make a system that helps generate the question paper with very ease. It offers to select questions randomly from the question bank, i.e. added by the instructor prior to this action for each course of different programs/courses and topics for different difficulty levels.

The end goal for the product is to come up with the system which can construct the question paper (pdf file) that can be printed or saved.

This proposed system can be used by any authorized users. It does not require much of technical knowledge, experience or expertise however basic computer operating knowledge to browse through websites is expected.
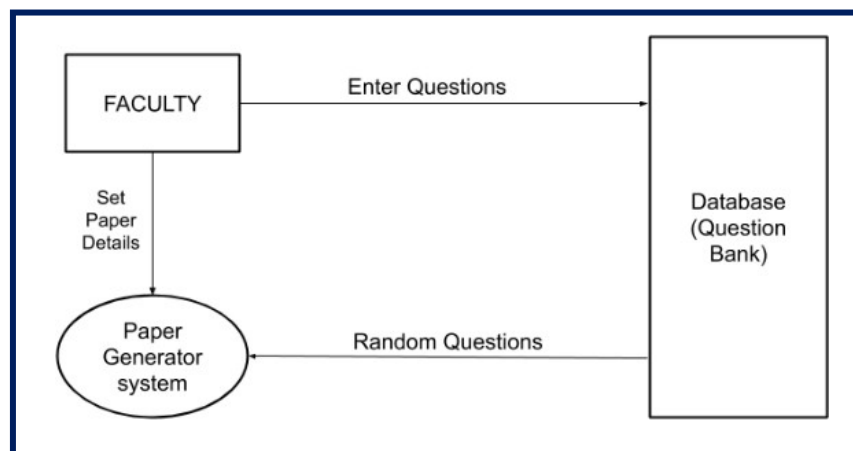
# 2. SOFTWARE REQUIREMENT SPECIFICATION

This section covers functional and non-functional requirements for the product. The primary purpose is to provide crisp data points, basis on which the project is designed.

## 2.1 SYSTEM REQUIREMENT STUDY

### 1. Login Domain
    1.1. User must be authenticated before using the console
    1.2. For invalid password, validation failure is responded back
    1.3. Forgotten password functionality is intended for retrieval of this password
    1.4. Upon successful login, the user will be redirected to the dashboard
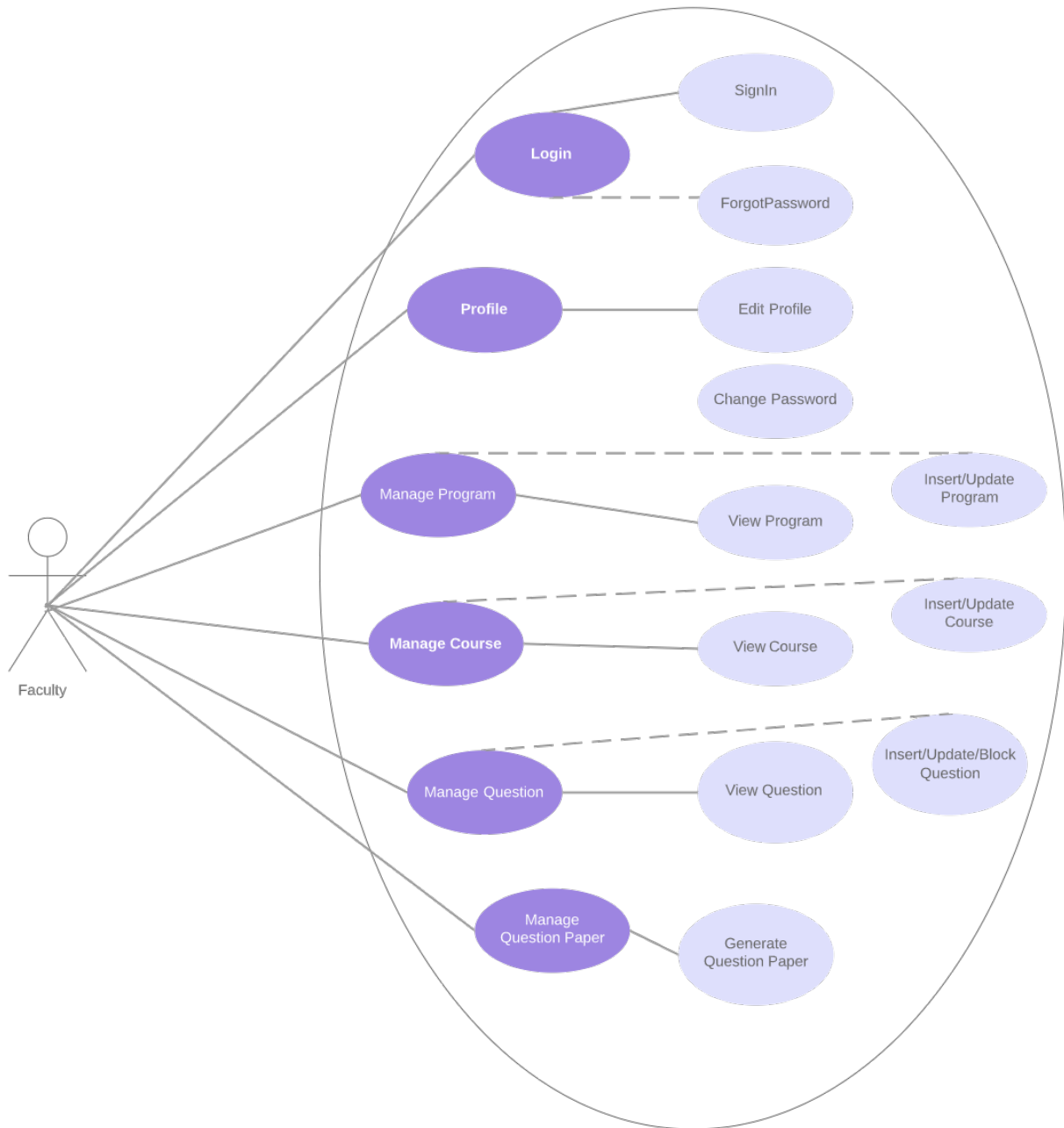    1.5. Session should be active infinitely post login

### 2. Faculty Domain
    2.1. Whenever any faculty logs in, he/she will be a provided dashboard allowing to navigate through an interface of course, program and question-paper
    2.2. Only active faculty can login
    2.3. Faculty can manage (insert, update, display) the course
    2.4. Faculty can manage (insert, update, display) the program
    2.5. Faculty can edit his/her profile
    2.6. Logged-in faculty can manage (Block and unblock) the other faculty
    2.7. Ability to manage the question bank (add| update | delete)
    2.8. Generate question paper based on marks and difficulty level
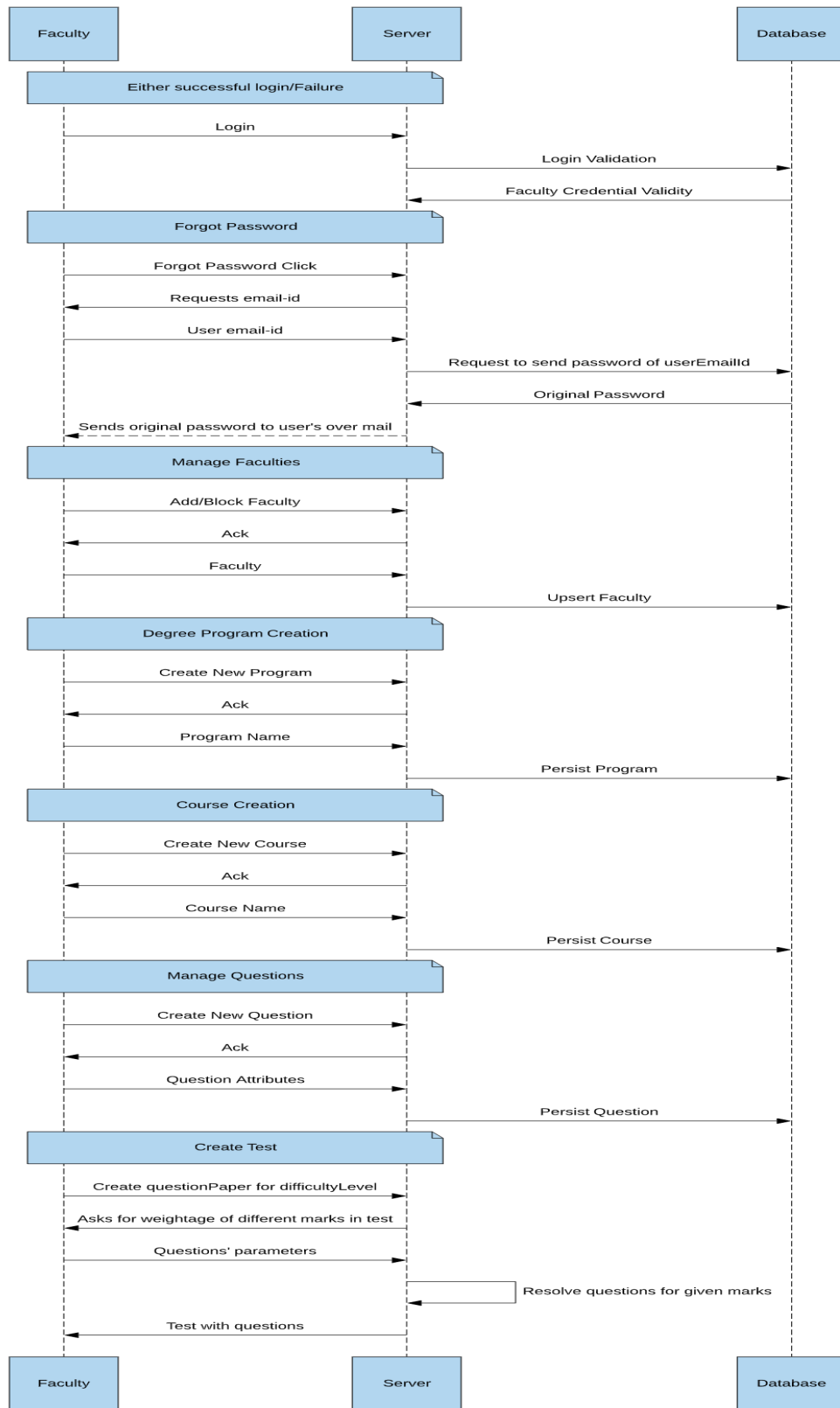    2.9. Download the question paper

## 2.2 TECHNICAL ARCHITECTURE

| | |
|---|---|
| **Tools** | **Draw.io, Visio, Visual Studio Code, MySQL(phpMyAdmin)** |
| **Operating System** | **Windows / MacOS** |
| **Language** | **Front-End: ReactJS, Bootstrap, CSS**<br>**Back-End: NodeJS (ExpressJS)** |

# 3. SYSTEM DESIGN

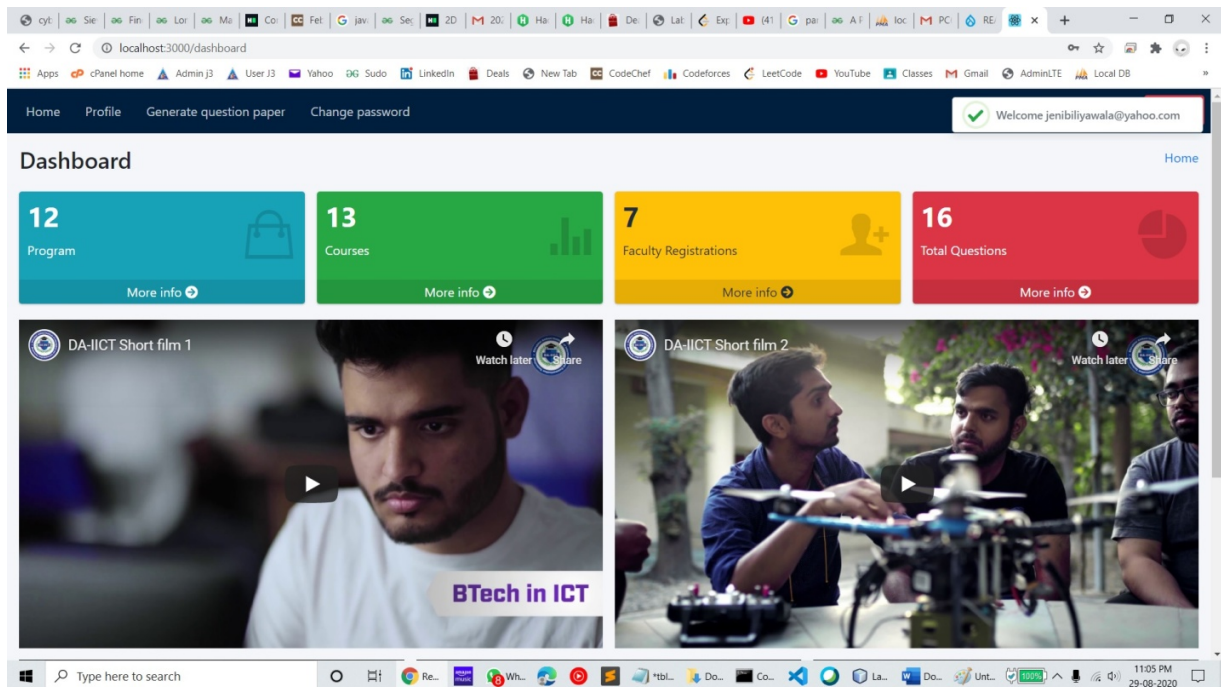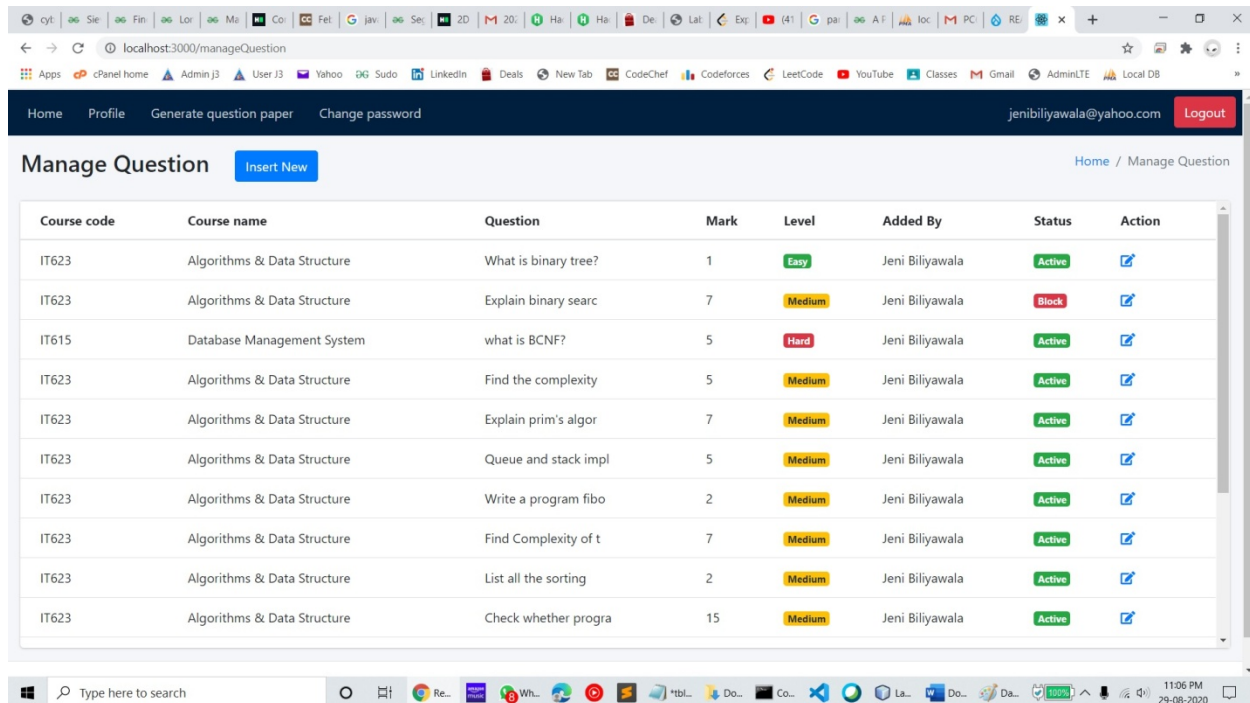## 3.1  USE CASE DIAGRAM

## 3.2 SEQUENCE DIAGRAM

**Faculty**      **Server**      **Database**

**Either successful login/Failure**

Login

Login Validation

Faculty Credential Validity

**Forgot Password**

Forgot Password Click

Requests email-id

User email-id

Request to send password of userEmailId

Original Password

Sends original password to user's over mail

**Manage Faculties**

Add/Block Faculty

Ack

Faculty

Upsert Faculty

**Degree Program Creation**

Create New Program

Ack

Program Name

Persist Program

**Course Creation**

Create New Course

Ack

Course Name

Persist Course

**Manage Questions**

Create New Question

Ack

Question Attributes

Persist Question

**Create Test**

Create questionPaper for difficultyLevel

Asks for weightage of different marks in test

Questions' parameters

Resolve questions for given marks

Test with questions

**Faculty**      **Server**      **Database**

# 4. SNAPSHOTS

- Login Page



- Home Page
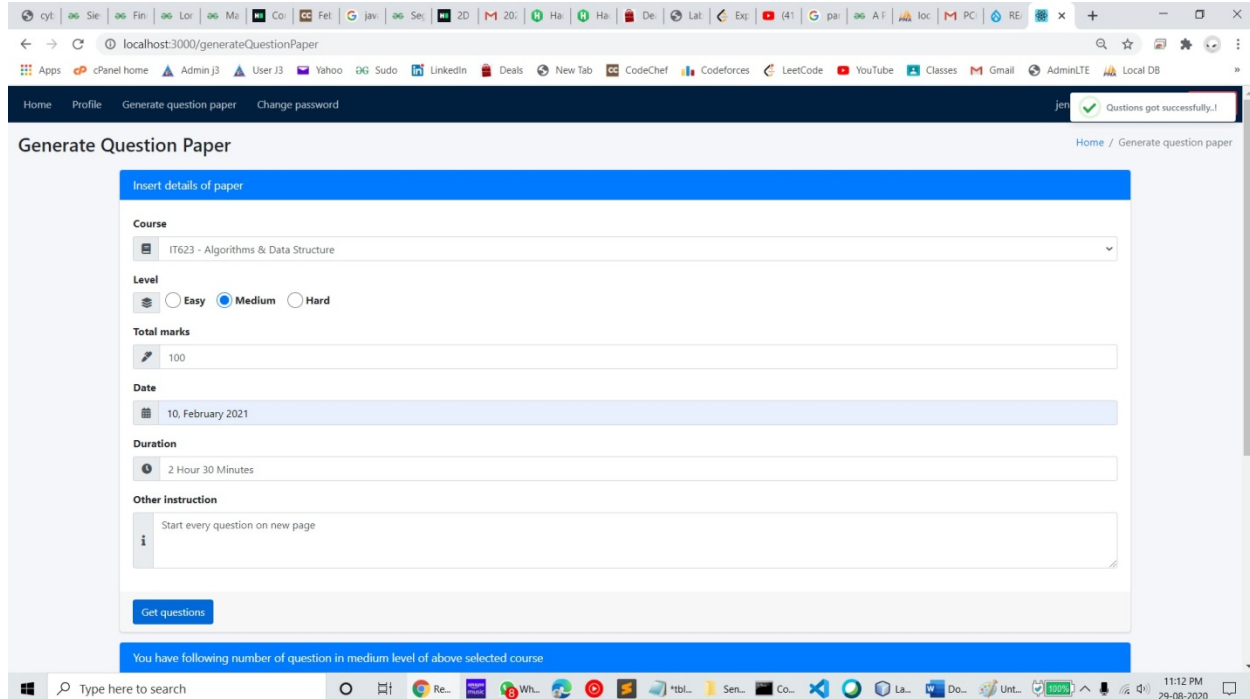
- Entity Display Page (Used for Question/Course/Faculty/Program Tabular Views)



- Entity Create/Update Form (Used for QuestionPaper generation & managing course/program/faculties)

# 5. TEAM MEMBER CONTRIBUTIONS & LEARNINGS

| TASK | JENI BILIYAWALA (***089) | VASANI VIBHUTI (***110) |
|------|:---:|:---:|
| Database Modeling | ✅ | |
| Login | ✅ | |
| Forgot Password | | ✅ |
| Password Reset | ✅ | |
| Faculty's Personal Profile | ✅ | |
| Manage Peer Faculties | ✅ | ✅ |
| Manage Program | ✅ | |
| Manage Course | | ✅ |
| Manage Question Bank | ✅ | |
| Generate Question Paper | ✅ | |
| Project UML | ✅ | |
| Project Documentation | | ✅ |

*Learning:*

❖ System designing concepts

❖ MVC architecture

❖ Latest tools and technologies, such as ReactJS, NodeJS, ExpressJS, React Troubleshooting Tools

❖ Responsive UI designs using open-source UI framework

❖ Team work

❖ Design documents

❖ Real life software development methodology – pros/cons (iterative waterfall)

# 6. REFERENCES

- React JS
- Node JS
- Express JS
- MySQL