

Online Academics: Requirements Document (version 1.75)

Project: Online Academics

Date(s): 5/06/21

Prepared by:

Jimmie Dale Reyes

Evan Delasota

Tuan Ngo

Document status: ☒ Draft ☐ Proposed ☐ Validated ☐ Approved

1. Introduction

This document contains the system requirements for **Online Academics**. These requirements have been derived from several sources, including MongoDB, Visual Studio, Nodejs, (More to be added)

1.1 Purpose of This Document

This document is intended to guide development of **Online Academics**. It will go through several stages during the course of the project:

1. **Draft:** The first version, or draft version, is compiled after requirements have been discovered, recorded, classified, and prioritized.
2. **Proposed:** The draft document is then proposed as a potential requirements specification for the project. The proposed document should be reviewed by several parties, who may comment on any requirements and any priorities, either to agree, to disagree, or to identify missing requirements. Readers include end-users, developers, project managers, and any other stakeholders. The document may be amended and repropose several times before moving to the next stage.
3. **Validated:** Once the various stakeholders have agreed to the requirements in the document, it is considered validated.
4. **Approved:** The validated document is accepted by representatives of each party of stakeholders as an appropriate statement of requirements for the project. The developers then use the requirements document as a guide to implementation and to check the progress of the project as it develops.

1.2 How to Use This Document

We expect that this document will be used by people with different skill sets. This section explains which parts of this document should be reviewed by various types of readers.

Types of Reader

Web developers and users who have some basic knowledge about web development can read this documentation.

Technical Background Required

The general background need are skills and understanding of:

- Visual Studio Code, JavaScript, HTML, CSS, SASS, Bootstrap for Front-end
- How to use package manager: NPM
- Basic knowledge of Linux OS
- Know about Nodejs, NOSQL, MongoDB and Express

1.3 Scope of the Product

A web app that allows professors to upload grades and students to view how well they are performing. Students can choose courses to add to their account. Students can add/delete/recover courses in their account page. We will be using Javascript, HTML, CSS, bootstrap for front end and Javascript, Nodejs, MongoDB for backend.

1.4 Business Case for the Product

This product purpose and goals are:

- An interactive learning web platform
- Must be easy to use
- A simple interface that shouldn't be cumbersome, yet is still aesthetically pleasing

1.5 Overview of the Requirements Document

- Must be working
- Must have an elegant design
- Must have good database
- Must prevent brute force
- Must be able to control easily

2. General Description

This section will give the reader an overview of the project, including why it was conceived, what it will do when complete, and the types of people we expect will use it. We also list constraints that were faced during development and assumptions we made about how we would proceed. A platform for teachers and students sharing free courses

2.1 Product Perspective

This product will be a simple way for students to check their grades and teachers to upload their pupils' performances. Other products may be more complicated or too cumbersome to use. The stakeholders are education institutions such as schools and colleges. The people who will benefit are the users, the students and teachers using the app.

2.2 Product Functions

- User can create their account
- Teachers can create/delete/change information of their courses
- Student can add/delete/recover their registered courses
- Student can select one or multiple courses to delete

2.3 User Characteristics

- Basic knowledge of using computer
- A student/teacher of a Educational Institution

2.4 General Constraints

- Teachers/Students/Anyone who wants to learn new subjects.

2.5 Assumptions and Dependencies

- Requires knowledge with javascript, HTML5, CSS, bootstrap
- Front End Development
- Back End Development
- Project Manager

3. Specific Requirements

This section of the document lists specific requirements for Online Academics Requirements are divided into the following sections:

1. User requirements. These are requirements written from the point of view of end users, usually expressed in narrative form.
2. System requirements. These are detailed specifications describing the functions the system must be capable of doing.
3. Interface requirements. These are requirements about the user interface, which may be expressed as a list, as a narrative, or as images of screen mock-ups.

3.1 User Requirements

- Computer system
- Network connection
- Keyboard
- Mouse
- Web browser

3.2 System Requirements

- Server: OS can run Nodejs and internet connection
- using NOSQL (MongoDB)
- Mongoddb account

3.3 Interface Requirements

- *Clear and simple*
- *Straightforward*
- *Compatible with any browse*

4. Appendices

Specified Scenarios:

Educational Institutions
Students
Teachers

Requirements:

MongoDB Account
Web Browser (Google Chrome)

Similar products:

TITANium

5. Glossary

Include a glossary of definitions, acronyms, and abbreviations that might be unfamiliar to some readers, especially technical terms that may not be understood by end-users or domain-specific terms that might not be familiar to developers.

Front-end: The coding that deals with the surface of the program.. This mostly deals with the aesthetics and what can be seen.

Back-end: The coding that happens in the background. This part mostly deals with what is not seen and what occurs in the background of the program.

Controllers: Controls and directs where the user will be directed when clicking on links or tabs.

Model: The description of the course, along with the name, picture associated, date of creation, etc.

Public/Img: Images and logos are stored here.

routes: The pathing of the site

views: The front-end visual aesthetic, coding done in Handlebars language.

6. References

Nodejs: <https://nodejs.org/en/>

Visual Code: <https://code.visualstudio.com/>

MongoDB: <https://www.mongodb.com/cloud/atlas>

SCSS: <https://sass-lang.com/documentation>

Handlebars: <https://handlebarsjs.com/guide/#what-is-handlebars>