horizontal line

Software Development Plan (CoursK)

**April 6, 2024**

Version 1.0

**Presented To:**

Dr-Mohamed Ramadan

**Submitted By:**

Mohamed Mostafa Ahmed Eltaieb, Mohamed Khalid Mohamed, Mohamed Elsayed Youssef,Kirolos Raouf Helmy.

# **REVISION HISTORY**

# Initial draft of project documentation.

[**REVISION HISTORY**](#_3znysh7) **1**

[**PRODUCT DESCRIPTION**](#_tyjcwt) **3**

[**2. TEAM DESCRIPTION**](#_3dy6vkm) **3**

[**3. SOFTWARE PROCESS MODEL DESCRIPTION**](#_3as4poj) **4**

[**4. PRODUCT DEFINITION**](#_1t3h5sf) **5**

[Context Diagram](#_4d34og8) 5

[Use Cases (click for individual descriptions)](#_2s8eyo1) 6

[Matrix of Responsibilities](#_lnxbz9) 12

[**6. PERT Chart**](#_1ksv4uv) **12**

[Test Strategy](#_44sinio) 13

[Wireframes](#_2jxsxqh) 13

[**8. RISK ASSESSMENT**](#_3j2qqm3) **14**

[Risk Identification](#_1y810tw) 14

[Risk Prioritization](#_4i7ojhp) 14

[Risk Mitigation](#_1pxezwc) 14

[**9. CONFIGURATION AND VERSION CONTROL**](#_2xcytpi) **15**

[**10. TOOLS**](#_1ci93xb) **15**

[**11. ARCHITECTURE**](#_3whwml4) **15**

[**\*\*\* Check hyperlink for larger image**](#_qsh70q) **1**

# **PRODUCT DESCRIPTION**

The project aims to create a website that serves as a centralized platform for free courses from various websites. Users can access a wide range of courses across different domains such as AI, ML, data science, cyber security, front end, Android , flutter, and backend. The website features a login page for users to access their accounts, and blocks representing different tracks. Each track includes available courses, a roadmap, instructor information, course duration, and ratings.

The website also incorporates a chatbot feature to assist users with their queries, providing a seamless learning experience. Overall, the platform aims to provide users with easy access to quality educational resources from multiple sources in one convenient location.

# **2. TEAM DESCRIPTION**

| Team Members    Concepts | Mohamed Eltaieb | Mohamed Khalid | Mohamed Elsayed | Kirolos Raouf |
| --- | --- | --- | --- | --- |
| Database Management |  |  | ☑ | ☑ |
| Front End |  | ☑ | ☑ |  |
| Chat bot | ☑ | ☑ |  |  |
| Data Mining | ☑ |  |  | ☑ |

The skills needed for this project are:

* Time Management
* Good Communication Skills
* Positive Attitude
* Interpersonal Skills
* Respect each others ideas
* Experience in programming
* Experience in database management
* Everyone is willing to learn

As a whole, our team is diverse in many concepts needed for this project.

**3. SOFTWARE PROCESS MODEL DESCRIPTION**

The chosen software process model for the project is Extreme Programming (XP). XP emphasizes customer satisfaction, rapid feedback, continuous testing, and close collaboration between team members. Pair programming will be utilized to enhance code quality and promote knowledge sharing among team members. Additionally, frequent releases and iterations will allow for quick adaptation to changing requirements and continuous improvement of the product.

# **4. PRODUCT DEFINITION**

**Functional Requirements:**

\*User Registration and Login:

-Users should be able to create an account by providing a username, password, Phone, and email address.

-Users should be able to log in using their registered credentials.

-Passwords should be securely hashed and stored in the database.

\*Course Categorization:

-Courses should be categorized into different domains such as AI, ML, data science, cybersecurity, front end, Android, Flutter, and backend.

\*Course Enrollment:

-Users should be able to enroll in courses they are interested in.

-Enrolled courses should be done by a link.

\*Chatbot Integration:

-The platform should integrate a chatbot feature to assist users with their queries.

-The chatbot should be able to answer common questions related to course info, etc.

-Users should have the option to interact with the chatbot through text input.

\*Admin Panel:

-An admin panel should be available for managing courses uploaded to the platform.

-Admins should be able to review course content.

-Admins should have the authority to approve or reject courses based on their suitability and effectiveness.

**Non-Functional Requirements:**

\*Performance:

-The website should load quickly and respond to user interactions without significant delays.

-Response times should be minimal, even under heavy user traffic.

-The platform should be able to handle concurrent user requests efficiently without crashing or slowing down.

\*Security:

-User data should be encrypted and securely stored in the database.

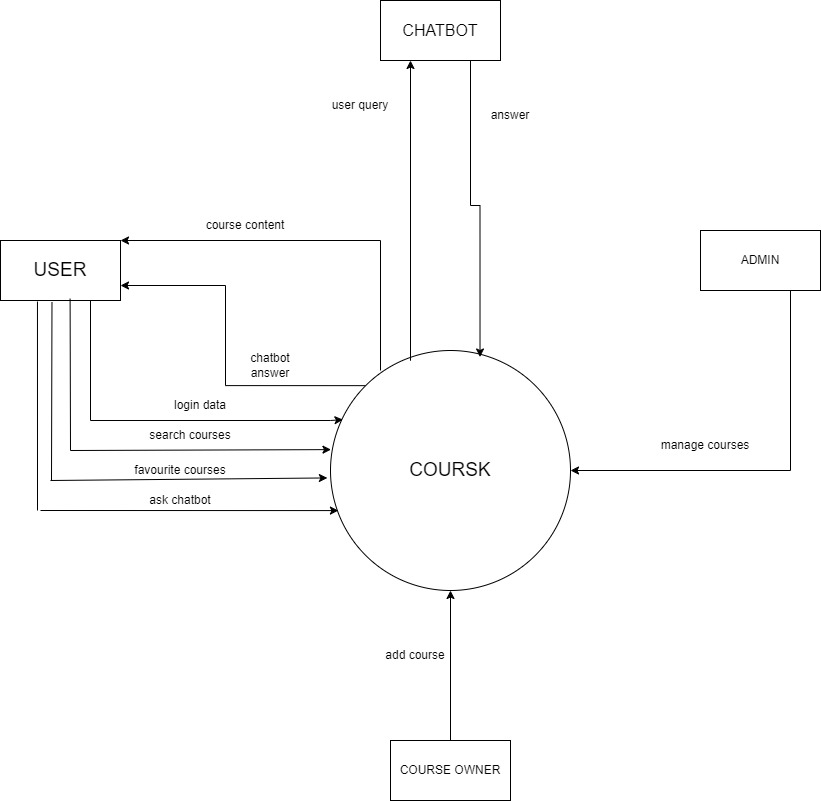
\*Usability:

-The website should have an intuitive and user-friendly interface.

-Navigation should be clear and intuitive, allowing users to easily find and access desired features.

-User interactions should be consistent across different pages and devices, ensuring a seamless user experience.

## **Context Diagram**



**Course Owners:**

The course owners are the people who add there courses to the platform. They login with a special account that requires there name and Password , register by name, email address, phone number and Password. This account is with limited authority to update on the website and the database.

**Users:**

The users are the students or anyone who wants to learn something in the provided courses. They can create an account with a username, phone number, and password and they need to input an email. These users are able to select a course, search for a specific one.Users also can use the chatbot provided by using prompts to get the desired course in a fast and easy way.

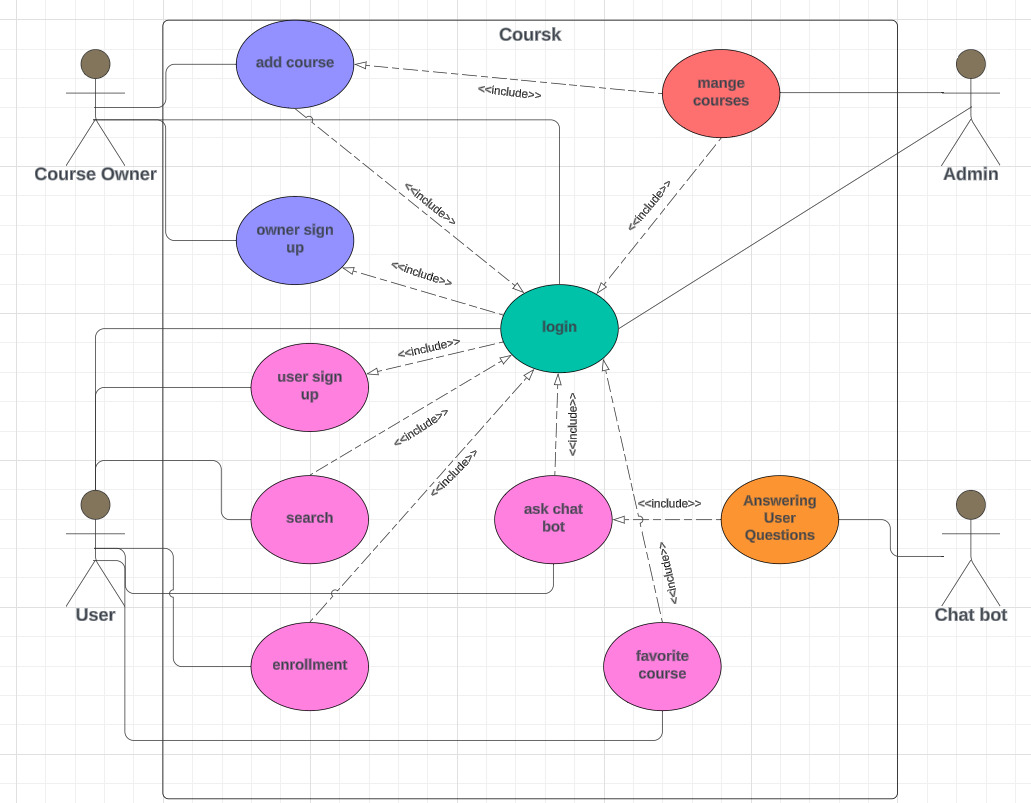
**Admin:**

He is the one who is responsible for managing the courses uploaded to the platform by reviewing there content and see if it’s suitable and effective or not.

**Chatbot:**

And AI assistant for the students over the platform

## [**Use Cases**](https://docs.google.com/document/d/12XgfNrVNvVZUDHk7BYBtv-j7iTASqcnU_EvzfFDVBv4/edit) **(click for individual descriptions)**



**Use Case #1.1:**

**Name:** User Sign up

**Participating Actor(s):** user

**Entry Conditions:** online course website is accessed

**Exit Conditions:** A new user account is created

**Flow of Events:**

1. user sign up as a user
2. enters Data
3. Data is Validated
4. New Account is created

**Special Requirements:** If data is invalid, display error message and try again

**Use Case #1.2:**

**Name:** course Owner Sign up

**Participating Actor(s):** Owners

**Entry Conditions:** online course website is accessed

**Exit Conditions:** A new owner account is created

**Flow of Events:**

1. Owner sign up as course owner
2. Enters Data
3. Data is validated
4. New account is created

**Special Requirements:** If data is invalid, display error message and try again

**User Case #2:**

**Name:** Login

**Participating Actor(s):** Users and owners

**Entry Conditions:** Account exists

**Exit Conditions:** User has access to their account

**Flow of Events:**

1. User enters login info
2. Login info is validated
3. User is sent to home page

**Special Requirements:** If account is invalid, show error message and try again

**Use Case #3:**

**Name:** *enrollment*

**Participating Actor(s):** user

**Entry Conditions:** User is logged in to a user account

**Exit Conditions:** user is enrolled in the course which he need it .

**Flow of Events:**

1. user choose the course
2. user click on enroll button
3. user is moved to the page to enroll on the platform of the course provided

**Special Requirements:** No special conditions

**Use Case #4:**

**Name:** *ask chat bot*

**Participating Actor(s):** user

**Entry Conditions:** User is logged in to a user account

**Exit Conditions:** chat bot answers user questions

**Flow of Events:**

1. user click on chat bot icon
2. user enters what he is asking about

**Special Requirements:** No special conditions

**Use Case #5:**

**Name:** Add course

**Participating Actor(s):** course owners

**Entry Conditions:** Owner is logged in to a course owner account

**Exit Conditions:** A New course is added

**Flow of Events:**

1. owner click on add a new course
2. owner choose the category of course
3. owner can add the course

**Special Requirements:** No special conditions

**Use Case #6:**

**Name:** Answering User Questions

**Participating Actor(s):** chat bot

**Entry Conditions:** A user submits a question

**Exit Conditions:** The user's question is answered successfully

**Flow of Events:**

1. understand questions
2. make queries
3. provide answer

**Special Requirements:** No special conditions

**Use Case #7:**

**Name:** mange course

**Participating Actor(s):** Admin

**Entry Conditions:** owners request to add courses to the website

**Exit Conditions:** the courses is added are suitable and effective.

**Flow of Events:**

1. if the course is suitable → confirm it and display it successfully on the platform
2. if the course is not suitable → the admin will not display it on the platform and inform the owner the reason for the rejection

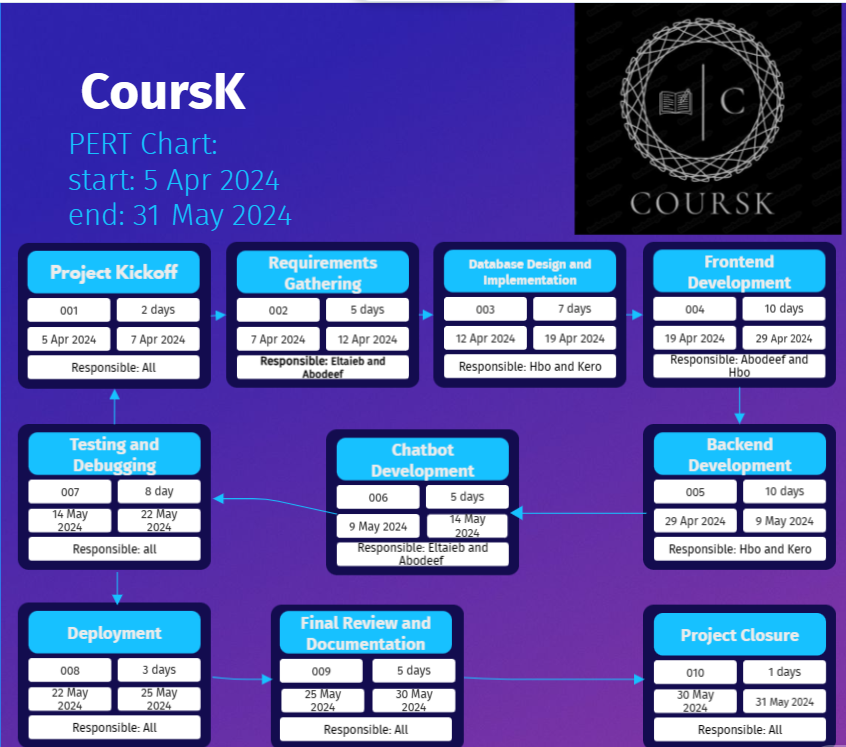
**Special Requirements:** No special conditions

**5. PROJECT ORGANIZATION**

## **Matrix of Responsibilities**

| Team Members    Concepts | Mohamed Eltaieb | Mohamed Khalid | Mohamed Elsayed | Kirolos Raouf |
| --- | --- | --- | --- | --- |
| Database Management |  |  | ☑ | ☑ |
| Front End |  | ☑ | ☑ |  |
| Chat bot | ☑ |  |  |  |
| Data Mining | ☑ |  |  | ☑ |

# **6.** [**PERT Chart**](https://drive.google.com/open?id=1rj1XyvQ63gfzoeDCI-cp3nwjir7VcBzN)



**7. VALIDATION PLAN**

## **Test Strategy**

* Unit testing: Developers will write unit tests for individual components and functions to ensure code correctness.
* Integration testing: Integration tests will be conducted to verify the interaction between different modules and components.
* User acceptance testing (UAT): Beta testing will be conducted with a group of users to gather feedback and identify any usability issues or bugs.
* Performance testing: Load testing will be performed to assess the website's performance under various traffic conditions.

## **Wireframes**

## 

# **8. RISK ASSESSMENT**

## **Risk Identification**

1-Technical challenges in integrating courses from various websites.  
2-Potential issues with data mining and chatbot implementation.   
3-Delays in backend development and database management.   
4-Inadequate testing leading to bugs and errors.

## **Risk Prioritization**

1. Technical integration challenges.
2. Backend development delays.
3. Data mining and chatbot implementation issues.
4. Testing inadequacies.

**Risk Mitigation**

-Regular communication and collaboration among team members to address technical challenges.

- Continuous monitoring and progress tracking to identify and address delays in backend development.

-Collaboration between team members responsible for data mining and chatbot implementation to mitigate potential issues.

-Thorough testing at each stage of development to identify and resolve bugs and errors promptly.

# **9. CONFIGURATION AND VERSION CONTROL**

Configuration and version control will be managed using Git and GitHub. The project repository will be hosted on GitHub, allowing team members to collaborate, track changes, and manage project versions effectively. Continuous integration practices will be implemented to ensure code quality and streamline the development process.

# **10. TOOLS**

The following tools will be utilized during the project:

* Programming Languages: HTML, CSS, JavaScript
* Database: sql (PHPMyAdmin)
* Version Control: GitHub
* Project Management: Microsoft Project
* Communication: Discord

# **11. ARCHITECTURE**

* The architecture of the project will follow a client-server model, with the frontend and backend components communicating over HTTP protocols. The frontend will be developed using HTML, CSS, and JavaScript to create dynamic and interactive user interfaces. On the backend, SQL (PHPMyAdmin) will be used as the database management system to store and retrieve data efficiently.

# **12-Deployment Plan:**

# **\*Install necessary dependencies and libraries:**

# **-Ensure all necessary dependencies and libraries are installed on the server environment.**

# **-Install web server software and database management system as required by the application.**

# **-Set up any additional software or tools needed for the proper functioning of the website.**

# **\*Perform final testing before going live:**

# **-Conduct final testing to ensure the deployed website functions correctly and meets the specified requirements.**

# **-Test all website features including user registration, chatbot integration, and admin panel functionality.**

# **-Perform compatibility testing across different browsers and devices to ensure a consistent user experience.**

# **-Conduct load testing to assess the website's performance under various traffic conditions and identify any potential bottlenecks.**

# **13-Maintenance and Support:**

# **Provide user manuals or guides for using the platform (By asking the chatbot):**

# **Create comprehensive user manuals or guides explaining how to navigate the website, register an account, Look for courses, use the chatbot, and access the admin panel.**

# 

# **Provide contact information for support and feedback:**

# **Set up dedicated channels for users to contact customer support or provide feedback, such as email support.**