

Author: R. Duisterhof

Product Description & Requirements | Version 1.2

PROOF OF CONCEPT

28 september 2024 | Project Name: MSCIP | Product name: Movie Scripter

potential domain: www.moviescript.io

Idea Description

The idea has been researched and concluded to be unique in the specific market of the music, books and film production industry. The general idea is to create an application that uses Artificial intelligence to IO E-books into Moviescripts.



General Setup

Backend Framework PHP: Laravel 10

JS Library: Vice, JQuery

CSS Framework: Bootstrap

Icon Library: mdi, font awesome, flaticons

Database: MySQL

Writing Method: CRUD + MVC

PMS: Trello

Version Control: Github [ReggieSBS/studios: Product](#)

[Studio CMS, CRM, ERP and AI Tools \(github.com\)](#)

AI Engine: ChatGPT 3.5 Turbo | model Whisper

INIT DEVOPS

The devops setup is based on the OTAP method.

Local Environment

Install a local host server for local development.

Check out branche: [ReggieSBS/studios: Product Studio CMS, CRM, ERP and AI Tools \(github.com\)](https://github.com/ReggieSBS/studios: Product Studio CMS, CRM, ERP and AI Tools)

Create a new branch with your name and ticket name

Clone dev branch

Configure Database file in laravel

Init Migrations

Push to personal branch

artisan serve & npm run dev

Test Environment

Merge personal branches into develop

Deploy develop branch on server domain dev.moviescript.io -> whitelisted environment

Acceptation Environment

Merge develop into test

Deploy test branch on server domain test.moviescript.io -> public accessible password protected

Production Environment

Minimal Requirements

The products can be marketed with the following functionalities for the end user to work

Login / Logout / Register

The possibility to Login / Logout and Register should be able by filling in a password (confirm the password) and entering an emailaddress which causes the system to send a validation email of their account by registration.

To increase experience their should be functionality to sign in or register with Google or Facebook.

User Interface Layout

The user interface should be attractive and visually aligned with the industry's aspects. this includes:

A Studio Type of interface that enables the user to:

Start at the studio dashboard and see a left menu and a top menubar.

1.1. The studio dashboard (landingspage after login)

should contain the following aspects:

1.1.a Load an e-book by file select or drag and drop

1.1.b See a loader change into a composer that tells the percentage of completion

1.1.c convert if so PDF into Textual content and display this into a textarea

1.1.d the functionality to either automatically or manually add chapters which are individually displayed as editable or lockable textareas and which are divided into pages and the pages should be carouselled by paginations functionality.

1.1.e The functionality to automatically detect book pages and chapters within the e-book

1.1.d The functionality to paginize the e-book in textarea textual content by click and radioboxes (2; a radiobox for paginize by chapter, and a radiobutton to paginize by e-book page)

1.2. The left menu should contain the following options:

Source Tree With + - functionality or a selectbox to see the movies

1.2.a. E-book Button (landingspage)

1.2.b. Pages Button

1.2.c. Chapters Button

1.2.d. Characters Button

1.2.e. Movie Script Button

1.2.e.x Questions

1.2.e.xx Archetypes

1.2.e.xxx Formula

1.2.e.xxxx Acts

- New Act

- Source Tree With Acts

1.3. The top menubar should contain the following options:

1.3.a left -> logo

1.3.b. right -> clickable profile picture with a dropdown menu

1.3.b.x Billing / Account

1.3.b.xx Profile Settings

1.3.b.xxx Logout

1.3.c. right -> help

1.3.d. right -> notifications and messages

1.3.e. right -> export

1.3.f middle -> clickable chatbot profile picture expended to ai or automatically opened when using AI functions within the views.

Views and functionalities

The user interface should be attractive and visually aligned with the industry's aspects. this includes:

Pages view

This view should contain the functionality to browse through the e-book page by page. It should have information about which page you're currently reading, editable text of the page from the book, a input field to define the chapter of the current page, the amount of words, approximate reading time, an play (read with audio text to speech button) and it should contain cards that contain text fields to input the characters mentioned on that page, An environment and ambiance description a generalized small description of the Acts for each character related to

that page and the archetype related. Acts related are defined at the chapter view and archetypes are defined at the acts details view.

Chapter view

Basically the same functionality only this view should contain the functionality to browse through the e-book chapter by chapter. It should have information about which chapter you're currently reading, editable text of the chapter from the book, a input field to define the chapter by name and number, the amount of words, approximate reading time, an play (read with audio text to speech button) and it should contain cards that contain text fields to input the characters mentioned on that chapter, An environment and ambiance description a generalized small description of the Acts -> which the user have to be able to choose which are related to the chapter. for each character related to that chapter the current archetype should be loaded and related by loading the acts and for the acts the user should be able to relate them to the chapter in the view.

Both of these pages should contain a chatbot integrated AI which can be asked to analyze and compartmentalize the current chapter or page and fill the cards content text areas automatically such as: summarize the characters of that specific context, and suggest the best description of the environment and ambience. The user should also be able to let the AI define which archetype of character / Act is best suitable to relate to that specific chapter or page -> but this functionality should be trained which is later discussed in this document.

Character view

The character view should contain information about the characters life cycle during the movie script meaning all the characters should be clickable. the detail page should contain a timeline or archetypes of the character and acts, and the main information according Contour (software) methodology is preset by default (available textareas and cards) there is ability to add a new card with a title that defines the type of content it holds in relation to the character. It should be that you can delete and add a new character.

Movie Script main view

This view should contain a timeline of the movie with all aspects (questions, characters, chapters, pages, archetypes, acts, chronologically displayed and clickable to add)

The subviews of the movie script menu button details out the aspects of the movie scripts and are CRUD

General View Functionality

- Media library containing sound files, image files and video files
- When help is activated a right class -> menu section with a general tutorial description of the main purpose of the view and the user functionality / workflow
-

Project Management

In general the whole project is realized with a wysiwyg point of view. Meaning we make it visible before we make it functional. If it's determined to make it functional to be able to make it visible we will write the functionality. This is because it seems that the efficiency in project development is increased by visible results which means that it's a motivator to see and touch the work which may or may not work yet, to get a feeling and will to finish it.

The functionality should be written with a CRUD method frontend. The visual result should be first created, read and deletable before editable.

The functionality MVC should be first defined and made visible before created meaning V before M before C.

Project Management Method is Scrum

Sprint are defined in trello or jira by tickets with focus on the above basic fundamentals of chronological development methodology. meaning every sprint contains the same workflow:

1. Template, menu, blades and general frontend functionalities that relates to every blade frontend such as dropdowns and routes -> the blade(s) chosen to work on defines the following function creation steps during the sprints
2. Models and Middlewares
3. Controllers back to Views
4. Test

After every sprint we merge every personal ticket branche to develop branche and deploy to test

Test Cases are based on the above minimal requirement of the sprints chosen.

Negative results are placed in the refinement backlog of PMS.

Weekly sprints will result in a 4th week backlog refinement sprint.

Acceptation is a 5-10 use cases of a potential leads which will be offered to use the application for free after finishing the minimal requirements of the development process. These are asked for feedback through forms.

If findings are easy to use, beautiful to look at and fast to get result, we go into production and launch the marketing strategy

GDPR Data Distribution and ISO compatibility

The whole application should be according european standards meaning:

secure socket layers

whitelisting if necessary

OTAP and GDPR regulated (encrypted personal data etc.)

2factor authentication enabling functionality

Monthly ended or change subscription

Full control over personal information within the platform

General terms of platform usability

Cookie notification and control

registers of data leaks

yearly pentests on servers

yearly pentest in applications

Login logout log

server log

action and event log

This will enable the compatibility to implement the software by ISO regulated companies such as 9001 and 27005.

Database Structure

Tabel	
acts	
archetypes	
archetypes_acts	
chapters	
characters	
ebooks	
failed_jobs	
licences	
media_library	
messages	
migrations	
movies	
notifications	
pages	
password_resets	
payments	
personal_access_tokens	
plots	
plot_roles	
users	
user_subscriptions	
21 tabellen	9