Flix2You Presentation

State College, PA 16801
903-525-3282

Pluto Data, Inc.

Andrew Goldner (atg5184@psu.edu)
Lacey Goodman (leg5505@psu.edu)
Gabriel Nwauche (gmn5192@psu.edu)

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Project Goal

Digital transformation has accelerated exponentially over the last couple of years. The consumer is more remote than ever, requiring high data and network architecture reliability. Our skilled team can perform the discovery, design, and implementation of a new database to meet this ever-growing demand.

Architecture

This proposal includes an analysis of the current database architecture, design, data mapping for the upgraded platform, and implementation of the new database. We will identify system requirements, permission sets, role responsibilities, legal issues and deploy both a development and production environment.

Modules

The module will include new database architecture, a dashboard view that provides a single pane of glass for administration, high availability, and disaster recovery. All tests are performed in a development environment before moving to production.

> *“Data is the most important element of your business today.”*

Outcome

The database updates will allow you to meet the demand of a growing market. A seamless transition to the new platform will limit downtown and risk while enabling growth and the ability to scale.

PROJECT OVERVIEW

Building a media content management platform to deliver a future-proof platform requires a complete overhaul of the existing infrastructure. A key component is modernizing a database accessed by twenty million users annually. Our company follows a proven methodology for success, including:

> ***"We want your data to work for you, and not the other way around."***

- Consultation
- Design and Review
- Project Management
- Implementation Development Environment
- User Acceptance Testing (UAT)
- Migrate to Production
- Change Management
- Closing Meeting

Your company has done an excellent job getting you to this point; now it's time to partner with an elite firm that sole focus is data lifecycle management.

PROJECT MANAGEMENT

Project Management Lead

Pluto Data will leverage its certified project management office (PMO) to manage and coordinate all tasks. We will follow a proven methodology from start to finish to discovery, design, planning, implementation, user acceptance testing (UAT), and steady-state support. Using Six Sigma practices, we have created a Gnatt graph and established a timeline.

- Discovery – In this phase, we will review the scope of work for the project, identify resources and review the timeline. A critical project element with a goal of transparency and expectation setting between our organizations. We will introduce the teams at a kick-off meeting and establish a line of communication. Project risks will also be discussed, and the course of action will be reviewed.
- Design Phase – Here, we will focus on the project schedule, specific tasks, dates of completion, and resources required. It’s critical to coordinate resources and understand scheduling for both parties properly. We will review the database architecture, network infrastructure, and system requirements. Changes to the design will need to happen in this phase before moving to planning. Billing terms, master agreements, and contract terms can be reviewed as required.
- Project Implementation – As we move to project implementation, we will monitor and control the process and resources using the latest secure collaboration tools. Communication reports will be generated weekly, along with a scheduled conference call. Issues will be addressed, and a resolution path will be established. The evolution of the project plan depends on solid execution in the implementation phase. We

understand this is critical to the success and mitigation of risk. Quality assurance is a priority, and in the event of a change control, we will follow the procedure established in our terms.

- User Acceptance Testing – During this period, our team and pilot group of users will test the system and all components. Ensuring a successful rollout to production from the development environment.
- Steady-State Support – The final launch to production and steady state support will occur in this phase. We will focus on lessons learned, review a project survey, and key performance indicators (KPIs). You will transition to our support team leveraging the software level agreement (SLA) to establish a line of communication in the future.

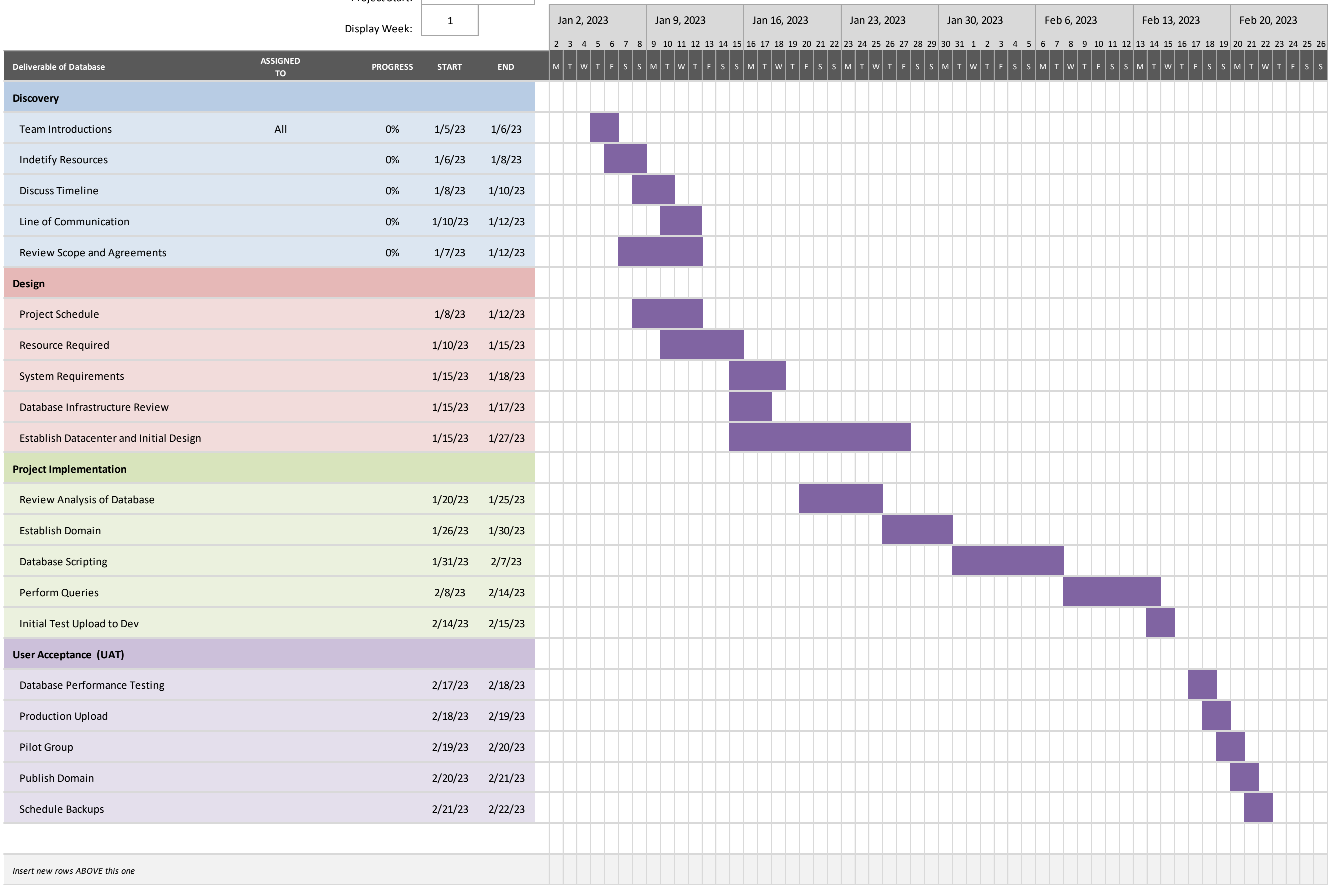
Project Charter Graph

Flix2You

Pluto Networks, Inc.
Project Lead - Andrew Goldner

Project Start: Thu, 1/5/2023

Display Week: 1



Deliverable of Database	ASSIGNED TO	PROGRESS	START	END
Discovery				
Team Introductions	All	0%	1/5/23	1/6/23
Indetify Resources		0%	1/6/23	1/8/23
Discuss Timeline		0%	1/8/23	1/10/23
Line of Communication		0%	1/10/23	1/12/23
Review Scope and Agreements		0%	1/7/23	1/12/23
Design				
Project Schedule			1/8/23	1/12/23
Resource Required			1/10/23	1/15/23
System Requirements			1/15/23	1/18/23
Database Infrastructure Review			1/15/23	1/17/23
Establish Datacenter and Initial Design			1/15/23	1/27/23
Project Implementation				
Review Analysis of Database			1/20/23	1/25/23
Establish Domain			1/26/23	1/30/23
Database Scripting			1/31/23	2/7/23
Perform Queries			2/8/23	2/14/23
Initial Test Upload to Dev			2/14/23	2/15/23
User Acceptance (UAT)				
Database Performance Testing			2/17/23	2/18/23
Production Upload			2/18/23	2/19/23
Pilot Group			2/19/23	2/20/23
Publish Domain			2/20/23	2/21/23
Schedule Backups			2/21/23	2/22/23
Insert new rows ABOVE this one				

Budget

QUANTITY	DESCRIPTION	UNIT PRICE	AMOUNT
16	Consulting	$185.00	**$3,034.00**
33	Professional Services Design	$225.00	**$7,380.00**
41	Project Management	$225.00	**$9,225.00**
164	Professional Services Implementation	$278.00	**$45,592.00**
3	Annual AWS Cloud Services	$195,000.00	**$585,000.00**
3	Annual Maintenance and Support	$9,118.40	**$27,355.20**
			$0.00

SUBTOTAL	**$677,586.20**
TAX RATE	**7.75%**
SALES TAX	**$52,512.93**
OTHER	
TOTAL	**$730,099.13**

In the event of a change of scope, this quote will follow the change control procedure.

Sign Below to Accept Quote:

Authorized Rep   Date

USER ANALYSIS

Role and responsibilities are crucial in designing and implementing any database structure. Privilege levels and permission sets must be established to determine rights level access. We will build these group policies from administrator to access level roles. Our basic user level focuses on the design and interface of the platform - the graphic user interface (GUI) and user experience modeling (UX). They will have read permissions to the data structure but cannot write against the schema.

The data analytics team will have read access to the entire scheme, with the ability to export content as a .csv or .xml to build reports and customized dashboards.

Developers and database engineers will have permission to edit and modify their code in a dev environment. With read/write access, they can test their queries and data models in an isolated environment.

Database admins will operate at the highest level and have access across tables and schemes to read, write, publish and delete data. They will work within an approval process for checks and balances. No single individual will have sole approval for executing data content.

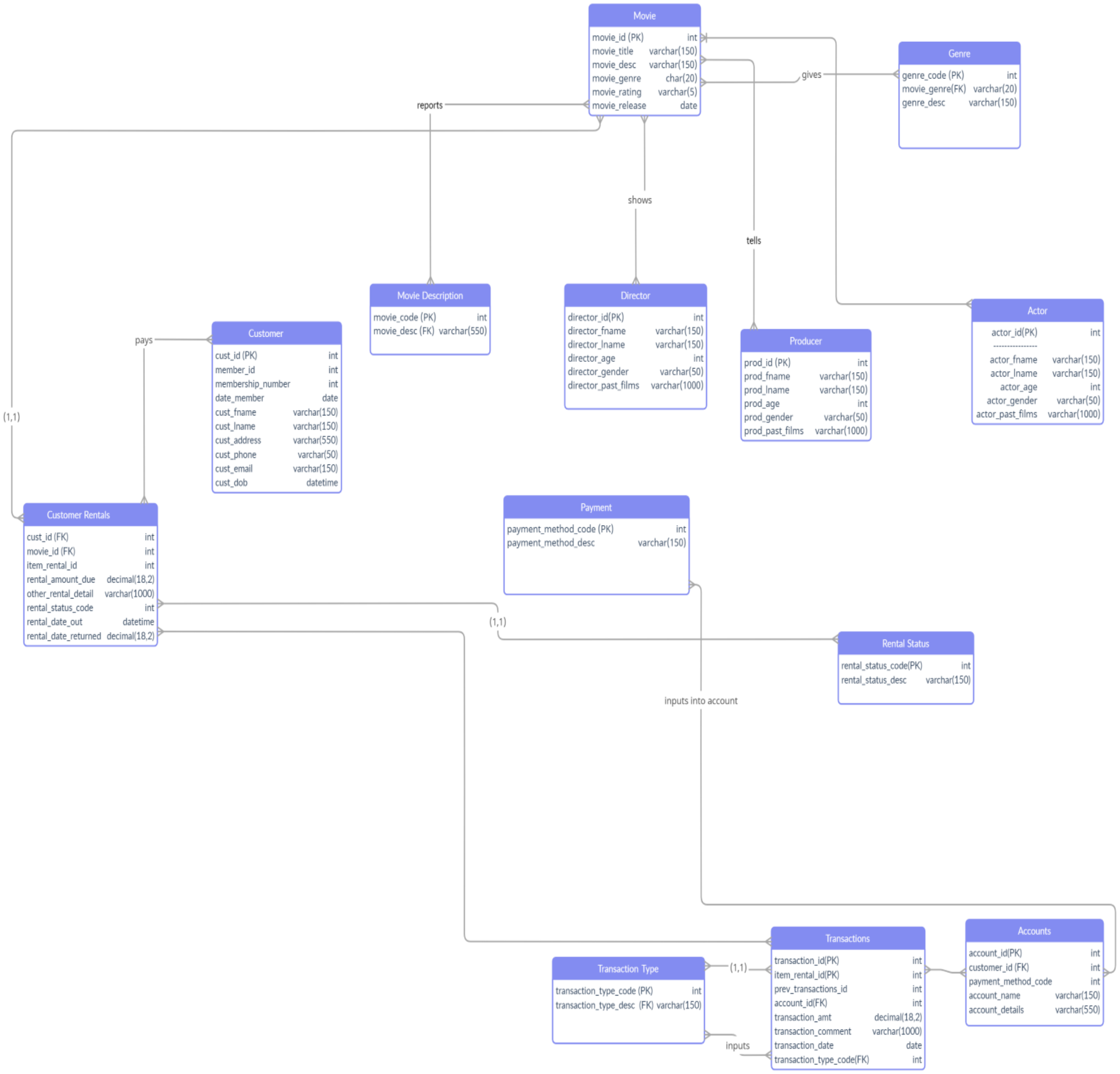
Examples of roles include:

- Db_dataux
- Db_dataanalytics
- Db_datadeveloper
- Db_dataengineer
- Db_dataadmin

ORGANIZATION DATA

Organization of Data

Statement
The role of the database structure is to neatly organize and distribute the data extracted from the queries, turning that said data into information. Our goal is to be able to have the database structure refined enough so we can pinpoint the exact object we need to when the time arises. The first thing we can do to clean up the database and make it a reliable option is to change the form from 1NF to 3NF



CREATE TABLE: TRANSACTION TYPE

Transaction_type	Transaction_type_code	transaction_type_desc

CREATE TABLE: PAYMENT METHOD

Payment_methods	Payment_method_code	Payment_method_description

CREATE TABLE: RENTAL STATUS

Rental_status	Rental_status_code	Rental_status_desc

CREATE TABLE: FORMS TRANSACTION

Forms_transactions	account_id	item_rental_id	prev_transaction_id	transaction_id	transaction_amt	transaction_comment	transaction_date	transaction_type_code

CREATE TABLE: ACCOUNTS

Accounts	account_id	cust_id	payment_method_code	account_name	account_details

CREATE TABLE: CUSTOMERS

Customers	cust_id	membership_number	date_number	cust_fname	cust_lname	cust_address	cust_phone	cust_email	cust_dob

CREATE TABLE: CUSTOMER RENTALS

Customer_rentals	cust_id	movie_id	item_rental_id	rental_amount_due	other_rental_detail	rental_status_code	rental_date_out	rental_date_returned	rental_amount_due

CREATE TABLE: MOVIE

Movie	movie_id	movie_title	movie_desc	movie_genre	movie_rating	movie_release

CREATE TABLE: MOVIE DESCRIPTION

Movie_description	movie_code	movie_desc

CREATE TABLE: GENRE

Genre	genre_codes	movie_genre	genre_desc

CREATE TABLE: DIRECTOR

Director	director_id	director_fname	director_lname	director_age	director_gender	director_past_films

CREATE TABLE: PRODUCER

Producer	prod_id	prod_lname	prod_age	prod_gender	prod_past_films

CREATE TABLE: ACTOR

Actor	actor_fname	actor_lname	actor_age	actor_gender	actor_past_films

```
TRANSACTIONS
CREATE TABLE transaction_types(
transaction_type_code int IDENTITY(1,1) NOT NULL,
transaction_type_desc varchar(150) NOT NULL);
-------------------------------------------------------------------------
PAYMENT
CREATE TABLE payment_methods(
payment_method_code int IDENTITY(1,1) NOT NULL,
payment_method_description varchar(150) NOT NULL);
-----------------------------------------------------------------------------
RENTAL STATUS
CREATE TABLE rental_status_codes(
rental_status_code int IDENTITY(1,1) NOT NULL,
rental_status_desc varchar(150) NOT NULL);
------------------------------------------------------------------------------
TRANSACTIONS
CREATE TABLE forms_transcations(
account_id int NOT NULL,
item_rental_id int NOT NULL,
prev_transaction_id int NOT NULL,
transaction_id int IDENTITY(1,1) NOT NULL,
transaction_amt decimal(18,2) NULL,
transaction_comment varchar(1000) NULL);
transaction_date date NULL,
transaction_type_code int NULL,
-------------------------------------------------------------------------------
ACCOUNTS
CREATE TABLE accounts(
account_id int IDENTITY(1,1) NOT NULL,
customer_id int not null,
payment_method_code int NOT NULL,
account_name varchar(150) NOT NULL,
account_details varchar(550) NOT NULL);
-------------------------------------------------------------------------------
CUSTOMERS
CREATE TABLE customers(
cust_id int IDENTITY(1,1) NOT NULL,
member_id int not null,
membership_number int NOT NULL,
date_member date`1 NOT NULL,
cust_fname varchar(150) NOT NULL,
cust_lname varchar(150) NOT NULL,
cust_address varchar(550) NOT NULL,
cust_phone varchar(50) NOT NULL,
cust_email varchar(150) NOT NULL,
cust_dob datetime NOT NULL);
------------------------------------------------------------------------
CREATE TABLE customer_rentals(
cust_id int NOT NULL,
movie_id int NOT NULL,
item_rental_id int IDENTITY(1,1) NOT NULL,
rental_amount_due decimal (18, 2) NULL,
other_rental_detail varchar (1000) NULL);
rental_status_code int NOT NULL,
rental_date_out datetime NOT NULL,
rental_date_returned datetime NULL,
rental_amount_due decimal (18, 2) NULL,
---------------------------------------------------------------------------
CREATE TABLE movie(
movie_id int IDENTITY(1,1) NOT NULL,
movie_title varchar(50) NOT NULL,
movie_desc varchar(550) NOT NULL,
movie_genre varchar(20) NOT NULL,
movie_rating varchar(5) NOT NULL,
movie_release date NULL;
-----------------------------------------------------
CREATE TABLE movie_description(
Movie_code int NOT NULL,
Movie_desc varchar(550) NOT NULL,
CREATE TABLE genre(
genre_code int NOT NULL,
movie_genre varchar(20) NOT NULL,
genre_desc varchar(150) NOT NULL,
---------------------------------------------------------
CREATE TABLE director(
Director_id int NOT NULL,
Director_fname varchar(50) NOT NULL,
Director_lname varchar(50) NOT NULL,
Director_age int NOT NULL,
Director_gender varchar(50) NOT NULL,
Director_past_films varchar(1000)  NULL,
---------------------------------------------------------------
CREATE TABLE producer(
Prod_id int IDENTITY (1,1) NOT NULL,
Prod_fname varchar(50) NOT NULL,
Prod_lname varchar(50) NOT NULL,
Prod_age int NOT NULL,
Prod_gender varchar(20) NULL
Prod_past_films varchar(1000) NOT NULL
--------------------------------------------------------------
CREATE TABLE actor(
Actor_fname varchar(50) NOT NULL,
Actor_lname varchar(50) NOT NULL,
Actor_age int NOT NULL,
Actor_gender varchar(20) NULL,
Actor_past_films varchar(1000) NOT NULL,

/*Primary Keys*/
ALTER TABLE customer_rentals ADD CONSTRAINT pk_item_rent_id PRIMARY KEY
(item_rent_id);
ALTER TABLE transaction_types ADD CONSTRAINT pk_transaction_type_code PRIMARY KEY
(transaction_type_code);
ALTER TABLE rental_status_codes ADD CONSTRAINT pk_rental_status_code PRIMARY KEY
(rental_status_code);
ALTER TABLE payment_methods ADD CONSTRAINT pk_payment_method_code PRIMARY KEY
(payment_method_code);
ALTER TABLE financial_transcations ADD CONSTRAINT pk_transaction_id PRIMARY KEY
(transaction_id);
ALTER TABLE accounts ADD CONSTRAINT pk_account_id PRIMARY KEY (account_id);
ALTER TABLE customers ADD CONSTRAINT pk_customer_id PRIMARY KEY (cust_id);
ALTER TABLE movie ADD CONSTRAINT pk_movie_id PRIMARY KEY (movie_id);
ALTER TABLE movie_description ADD CONSTRAINT pk_movie_code PRIMARY
KEY(movie_code);
ALTER TABLE genre ADD CONSTRAINT pk_genre_code PRIMARY KEY(genre_code);
ALTER TABLE director ADD CONSTRAINT pk_director_id PRIMARY KEY(director_id);
ALTER TABLE producer ADD CONSTRAINT pk_prod_id PRIMARY KEY (prod_id);
ALTER TABLE actor ADD CONSTRAINT pk_actor_id PRIMARY KEY(actor_id);
-------------------------------------------------------------------------------------------------------------------------
/*Foreign Keys*/
ALTER TABLE cust_rentals ADD CONSTRAINT fk_customer_rentals FOREIGN KEY(customer_id)
REFERENCES customers (customer_id);
ALTER TABLE customer_rentals ADD CONSTRAINT fk_movie_customer_rentals FOREIGN
KEY(movie_id)
REFERENCES movies (movie_id);
ALTER TABLE cust_rentals ADD CONSTRAINT fk_customer_rental_codes FOREIGN
KEY(rental_status_code)
REFERENCES rental_status_codes (rental_status_code);
ALTER TABLE financial_transcations ADD CONSTRAINT fk_transactions_account FOREIGN
KEY(account_id)
REFERENCES accounts (account_id);
```

```
ALTER TABLE financial_transcations ADD CONSTRAINT fk_transactions_rental FOREIGN
KEY(item_rental_id)
REFERENCES cust_rentals (item_rental_id);
ALTER TABLE financial_transcations ADD CONSTRAINT fk_transaction_previous FOREIGN
KEY(previous_transaction_id)
REFERENCES financial_transcations (transaction_id);
ALTER TABLE financial_transcations ADD CONSTRAINT fk_transaction_codes FOREIGN
KEY(transaction_type_code)
REFERENCES transaction_types (transaction_type_code);
ALTER TABLE accounts ADD CONSTRAINT fk_customer_accounts FOREIGN KEY(cust_id)
REFERENCES customers (cust_id);
ALTER TABLE accounts ADD CONSTRAINT fk_customer_payments FOREIGN
KEY(payment_method_code)
REFERENCES payment_methods (payment_method_code);
ALTER TABLE movie ADD CONSTRAINT fk_movie_code_movie FOREIGN KEY(movie_code)
REFERENCES movie(movie_code);
--------------------------------------------------------------------------------
/*Insert codes Script for 6 tables*/
Insert into genre_code values (‘Action’);
Insert into genre_code values (‘History’);
Insert into genre_code values (‘Horror’);
Insert genre_codes values(‘Sci-fi);
Insert genre_codes values(‘Romantic’);
Insert genre_code values (‘Drama’);
Insert genre_code values (‘Comedy’);
Insert genre_codes values(‘Seasonal’);
Insert genre_codes values (‘Thriller’);
Insert genre_codes values (‘Kids’)

Insert movie_title values (‘To Kill A Mockingbird’);
Insert movie_title values (‘The Revenant’);
Insert movie_title values (‘Silence Of The Lambs’);
Insert movie_title values (‘The Lion King’);
Insert movie_title values (‘Halloween’);
Insert movie_title values (Home Alone 2’);
Insert movie_title values (‘Happy Feet’);

Insert rental_date_out values (‘11-03-2019');
Insert rental_date_out values (‘03-12-2018');
Insert rental_date_out values (‘08-08-2013');
Insert rental_date_out values (‘09-21-2022');
Insert rental_date_out values (‘06-23-2011');
Insert rental_date_out values (‘08-19-2017');
Insert rental_date_out values (‘04-06-2020');

Insert cust_address values (‘288 Hyland Castle Road’ , ‘Chicago’, ‘IL’, ‘16738-18937’);
Insert cust_address values(‘6700 Receptor Lane’, ‘Ann Arbor’ , ‘MI’ , ‘48103-48113');
Insert cust_address values(‘26 Runfield Plaza’ , ‘Greenwich’ , ‘CT’ , ‘06787-06878’);
Insert cust_address values(‘9273 Culombo Court’, ‘Wyckoff’, ‘NJ’ , ‘07481’);

Insert rental_status_code(rental_status_desc) values (‘Rented’);
Insert rental_status_code(rental_status_desc) values(‘In Stock’);
Insert rental_status_code(rental_status_desc) values(‘Overdue!’);

Insert into Customer(cust_id, member_id, membership_number, date_member, cust_fname,
cust_lname, cust_address, cust_phone, cust_email, cust_dob) values (‘0784’ , 1620983, 746533,
GETDATE(), ‘Greg’, ‘Heron’, ‘237 Fisher Glendon Avenue, Madison, WI, 53558-53715',
‘Gregoryheron@aol.com’, ‘09-14-1995')
Insert into Customer(cust_id, member_id, membership_number, date_member, cust_fname,
cust_lname, cust_address, cust_phone, cust_email, cust_dob) values (‘8671’ , ‘1754120’, ‘836252’,
GETDATE(), ‘Mary’, ‘Ellis’, ‘7188 Rover Lane, Lexington, KY, ‘40503-40517', ‘Marilynellis@yahoo.com’,
‘08-27-2000');
/*End Script*/
```

As you can see, there was not much variation in the previous model as the one we are replacing it with. This makes it easier to specificize the point of data you are trying to transition into information (Knowledge). This example may seem convoluted to begin with but when put into the data it can make access easier.

DATABASE ADMINISTRATION

Statement

Database Administrator

The job of a Database Administrator is to ensure the growth of the database and to facilitate any necessary changes to the database. In ensuring the proper operation and maintenance of the database a database administrator will provide success to everyone in the company. The main tasks of a Database Administrator are

Database implementation
Database planning
Database Analysis
Database Design

Other Database users

System Analyst
Security System Administrator

Database Backup and Recovery

In order to better protect all the assembled information located within the database we will utilize the technique database mirroring.

Mirroring is a useful technique that will allow for there to be two copies of a database that is necessary to be placed on a different server. These servers are general located in a secondary location

- Availability of database is increased
- Improved Data Protection
- Expedited availability for improvising the Database

High- safety Mode

A session of mirroring a database that can be performed synchronously or with someone present.

- Equipped with an automatic failover
- If a witness present, they can support an automatic failover
- The principal and mirror server work in tandem when syncing to make the transaction as quick as possible

Resolving Technical issues

Establishing a process to check, test, and resolve any potential problems on the website
Using the High-Safety-Mode will allow for changes to be made with out halting the entire system
Having a physical and cloud storage system will allow for safer storage

Storage

Dell PowerEdge R630 powered by intel Xeon processor E5-2600 V3

This technology will allow for plenty of storage space and allow for room to grow. It will provide fast speeds for updates up to 17% faster than its competitors.

Amazon Web Storage is an excellent tool that is affordable for storage and SQL Server

- Amazon utilizes Amazon Aurora. This is a MySQL and PostgreSQL compatible engine

Data Access and Security

Amazon Web Storage allows for websites to strengthen their security for their clients and employees

- AWS IAM- Allows for user to be created and controlled permissions if needed.
- Amazon Guard Duty- actively looks for any potential malicious
- Security Hub- Helps support the industry standard in security
- AWS Secrets Manager- Finds and keeps sensitive information

The customer's data will be used for the company's internal marketing. It will allow us to understand and create future products and services to cater to our customers. We wish to create a list to fulfill the likes and dislikes of those the utilize our services.

Amazon Web Storage will provide all the necessary security that is needed to protect the customers information on the back end of the website. In order to provide a sense of security for the individual utilizing our services a privacy policy needs to be established and posted on the website for the public eye. We shall provide all available contact information so there is an open line of communication between us and our users.

Yes, employees should sign a privacy disclaimer to better protect the privacy of our customers.

Data load

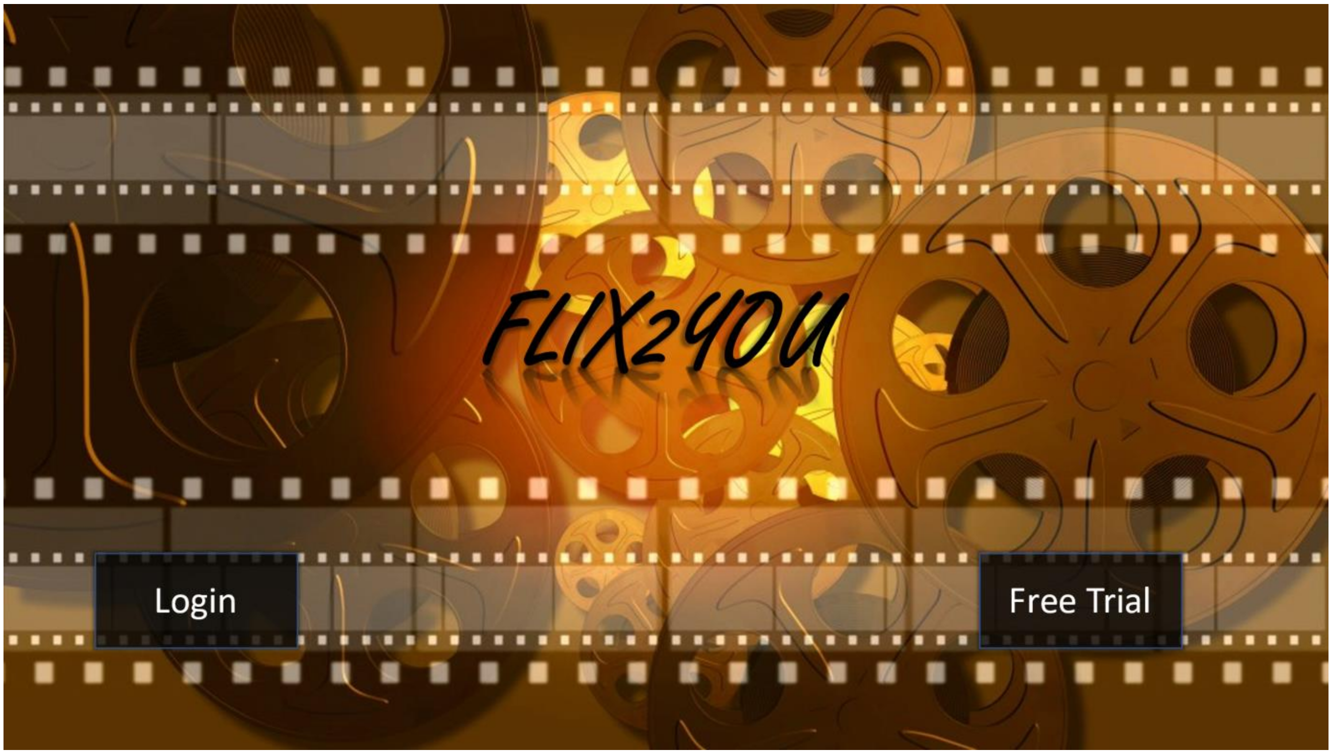
We will move forward with a Read Only database. This will allow us to move forward without having to add additional locks. In establishing more detailed indexes that will allow us to optimize our data retrieval without the breakdown of the index or any data mods. All updates will be held weekly early Wednesday morning between 2am and 3am.

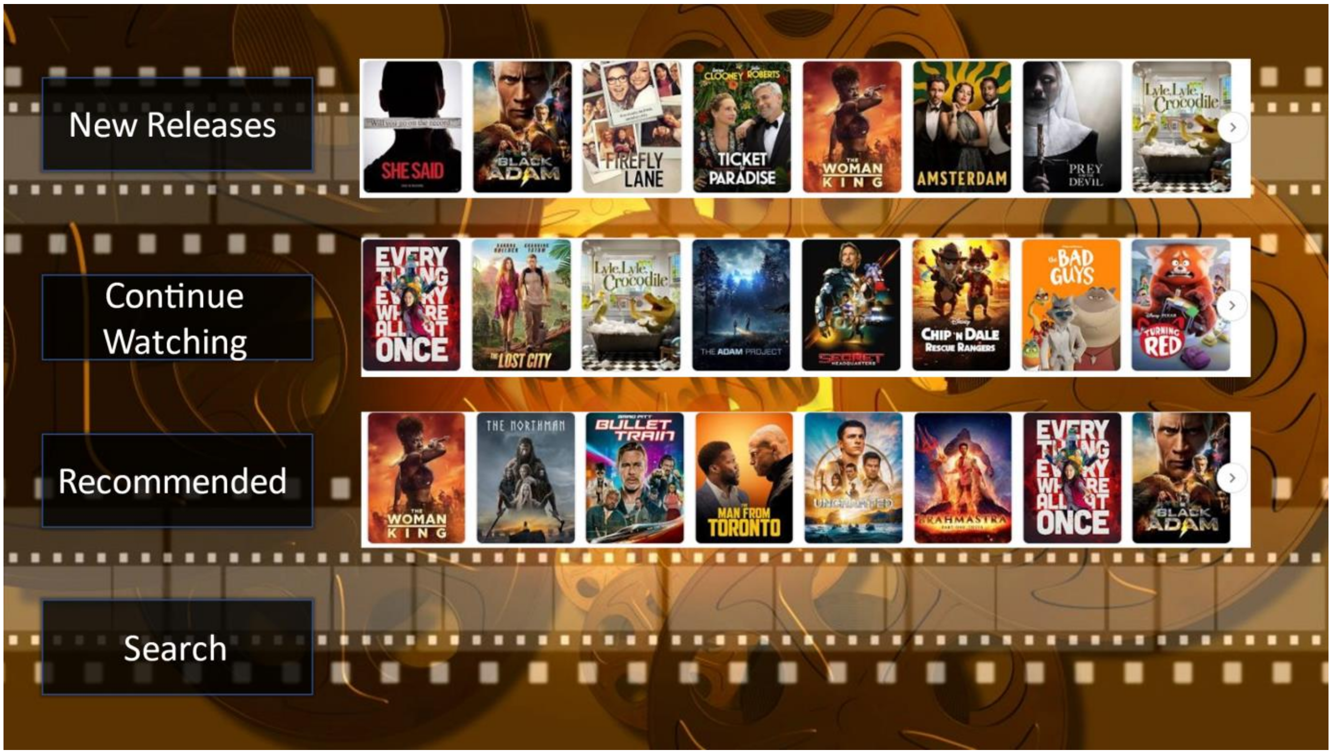
DATABASE DASHBOARD AND ANALYTICS

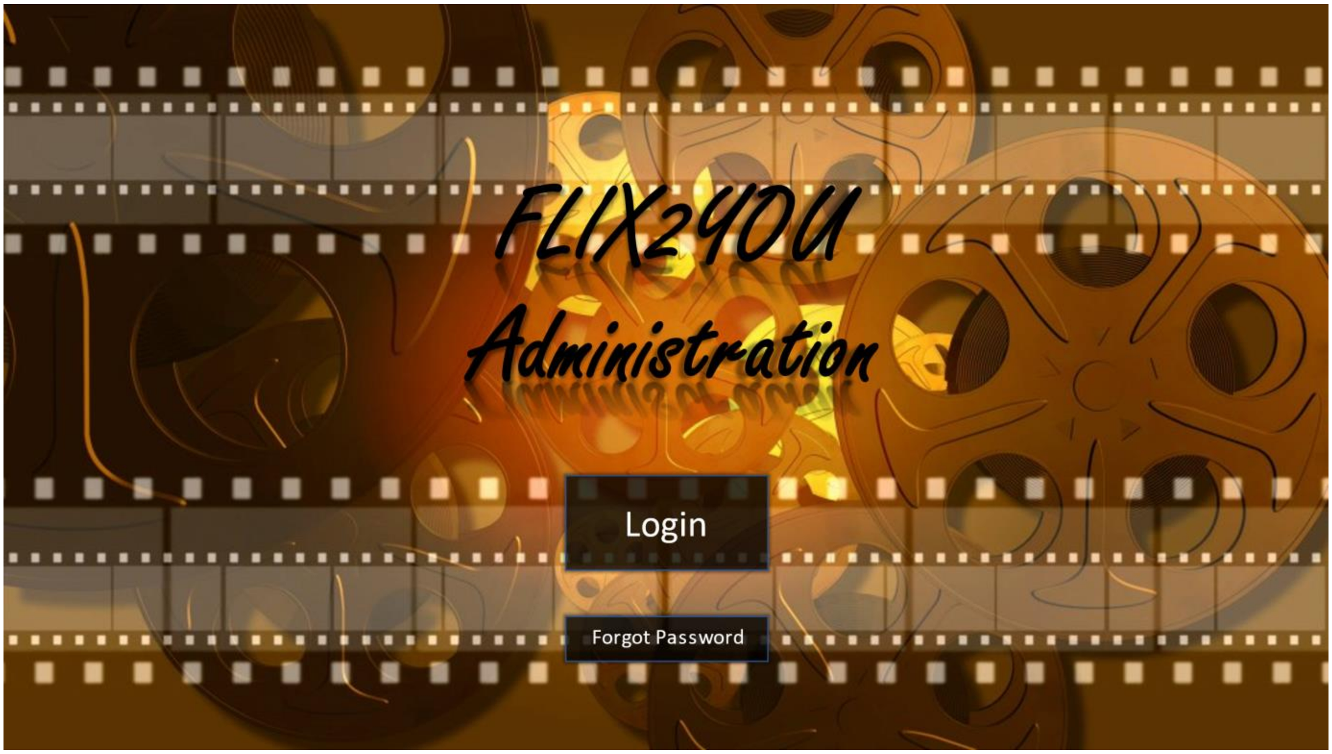
We will provide a user-friendly dashboard for subscribers with a focus on functional simplicity. Leveraging AI and Machine Learning will allow titles to be recommended based on user activity. Data will drive the analytics, and provide a sticky end user experience.

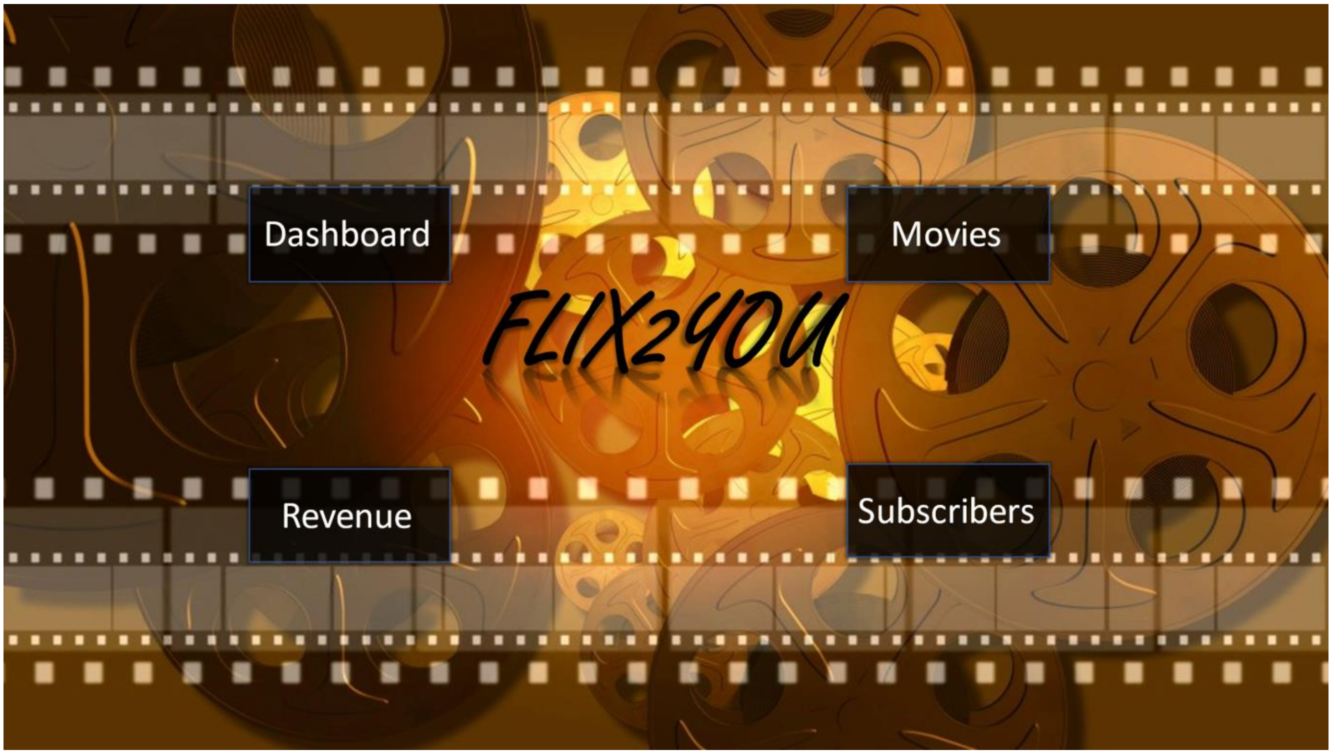
A second dashboard will be created for FLIX2YOU administrators, allowing for quick access to metrics and revenue trend. The landing page will be focus on a single pane of glass to provide the most relevant data at the forefront – thus enabling quick decision making.

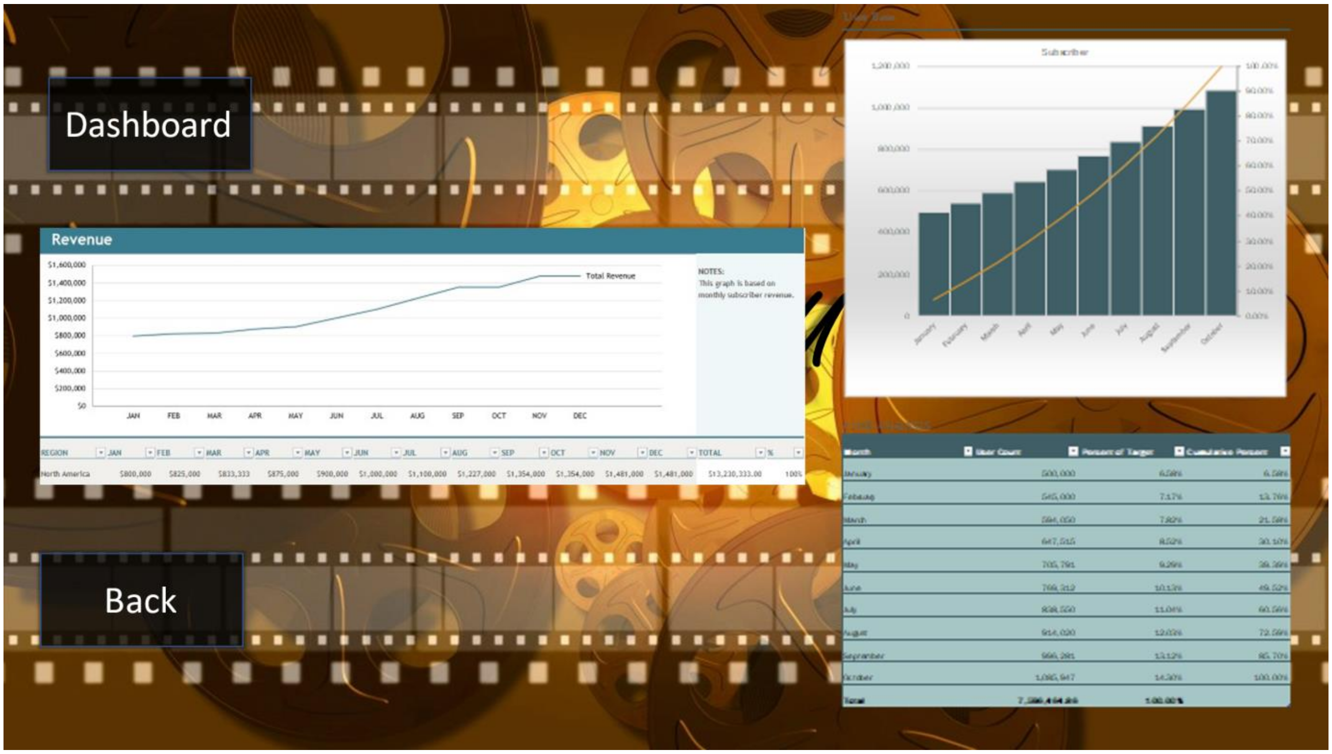
Example screenshots below:




New Releases
Continue Watching
Recommended
Search


FLIX2YOU
Administration
Login
Forgot Password





Revenue

NOTES:
This graph is based on monthly subscriber revenue.

REGION	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	TOTAL	%
North America	$800,000	$825,000	$833,333	$875,000	$900,000	$1,000,000	$1,100,000	$1,227,000	$1,354,000	$1,354,000	$1,481,000	$1,481,000	$13,230,333.00	100%

Month	User Count	Percent of Target	Cumulative Percent
January	500,000	6.58%	6.58%
Febuary	545,000	7.17%	13.76%
March	594,050	7.82%	21.58%
April	647,515	8.52%	30.10%
May	705,791	9.29%	39.39%
June	769,312	10.13%	49.52%
July	838,550	11.04%	60.56%
August	914,020	12.03%	72.59%
September	996,281	13.12%	85.70%
October	1,085,947	14.30%	100.00%
Total	7,596,466.86	100.00%	

LEGAL

We aim to provide the highest level of compliance, following all current regulations. Our team will work to establish group policies for users with rights-based privileges and access. All content will be archived in Azure Media Services Cloud, automatically encrypted using Advanced Encryption Standard (AES). AES is a symmetric encryption method that meets the highest that only privileged users can access for use by the US government. Flix2You will use a 256-bit key length for encryption with AES for maximum protection. A key length of 1024 bits is used to encrypt the symmetric keys. A new symmetric key is generated for each media file. Our employees follow strict company guidelines and must sign a non-disclosure agreement. They will adhere to our corporate policy guidelines when working with your data.

REFERENCES

https://www.geeksforgeeks.org/different-types-of-database-users/

https://learn.microsoft.com/en-us/sql/database-engine/database-mirroring/database-mirroring-sql-server?view=sql-server-ver16

https://www.simplilearn.com/systems-administrator-article

https://www.scylladb.com/glossary/database-administrator/

https://www.missioncloud.com/blog/the-top-10-security-tools-for-your-aws-environment

https://www.sitelock.com/blog/is-this-website-safe/

SIX SIGMA: A COMPLETE STEP-BY-STEP GUIDE July 2018 Edition
https://www.sixsigmacouncil.org/wp-content/uploads/2018/08/Six-Sigma-A-Complete-Step-by-Step-Guide.pdf

The IT Roadmap for Digital Business Transformation
https://emtemp.gcom.cloud/ngw/globalassets/en/information-technology/documents/insights/the-gartner-it-roadmap-for-digital-buisness-transformation-excerpt.pdf

https://azure.microsoft.com/en-us/pricing/calculator/

https://azure.microsoft.com/en-us/pricing/calculator/?service=media-services

Security White Paper August 2022
https://www.dashlane.com/download/whitepaper-en.pdf

https://aws.amazon.com/media-services/

https://aws.amazon.com/elastictranscoder/