**Online News Portal(S1-23)**

| Version Number | Date | Author/Owner | Description of Change |
| --- | --- | --- | --- |
| 1.0 | 30-July-23 | Shirsh Gupta |  |

1. **Problem Statement & Requirements Definition**

***ONLINE NEWS PORTAL:***

In today's digital age, the demand for timely and reliable news is ever-increasing. To meet this demand, we aim to develop an Online News Portal that provides users with access to the latest news articles across various categories, such as politics, entertainment, sports, and technology. The portal will also include user authentication features, allowing journalists and authors to submit and manage their articles on the platform.

**Requirement:**

*1. User Authentication:*

* Users should be able to create accounts and log in to the news portal securely.
* Implement password hashing and encryption to ensure the security of user credentials.
* Registered journalists and authors should have access to additional features, such as article submission and management.

*2. Frontend Interface:*

* Design an intuitive and user-friendly frontend interface for the news portal.
* Implement navigation elements, such as a navigation bar or sidebar, for easy access to different news categories.
* Display news articles with relevant details, including the title, category, author, and publication date.
* Include a search bar to allow users to search for specific articles based on titles or content.

*3. Backend Services:*

* Develop backend services to handle user authentication and authorization.
* Implement API endpoints to fetch and display news articles from the database.
* Create API endpoints for user registration, login, and article submission.

*4. Article Management:*

* Journalists and authors should have the ability to submit new articles through a dedicated form.
* Allow authors to edit and update their published articles as needed.
* Provide an interface for authors to view and manage their submitted articles.

*5. Database Integration:*

* Set up a database (e.g., SQLite) to store news articles and user information.
* Articles should be stored with relevant metadata, such as the title, content, category, author, and publication date.
* User accounts and authentication details should be securely stored in the database.

*6. Error Handling and Validation:*

* Implement robust error handling and validation mechanisms throughout the application.
* Validate user input to prevent potential security vulnerabilities and ensure data integrity.

*7. User Experience and Responsiveness:*

* Design the news portal to be responsive and accessible on various devices, including desktops, tablets, and mobile phones.
* Optimize page loading times for a smooth and seamless user experience.

*8. Security and Privacy:*

* Apply security best practices to protect user data and prevent unauthorized access to sensitive information.
* Ensure that only authenticated users can access certain features, such as article submission.

*9. Article Search and Filtering:*

* Implement search functionality to allow users to find articles based on keywords or categories.
* Provide options to filter articles by date, category, or author to enhance user navigation.

*10. Testing and Quality Assurance:*

* Conduct comprehensive testing, including unit testing, integration testing, and user acceptance testing, to ensure the application's functionality and reliability.
* Address and resolve any identified bugs or issues before deploying the news portal.

**Benefits:**

* A comprehensive online news platform that caters to users' information needs across various categories.
* User-friendly interfaces that enhance the user experience and engagement.
* Empowerment of journalists and authors to contribute their articles to a wider audience.
* Secure and authenticated access to ensure user data privacy and integrity.
* Efficient article management features for authors to maintain and update their content easily.

**Project Scope:**

The project scope includes the development of an online news portal with user authentication and article management features. The application will allow users to register, log in, view news articles, search for specific content, and allow journalists/authors to submit and manage their articles. The platform will be designed for responsiveness, scalability, and security, adhering to industry best practices and user requirements. The project will undergo rigorous testing and quality assurance to deliver a robust and reliable news portal for users.

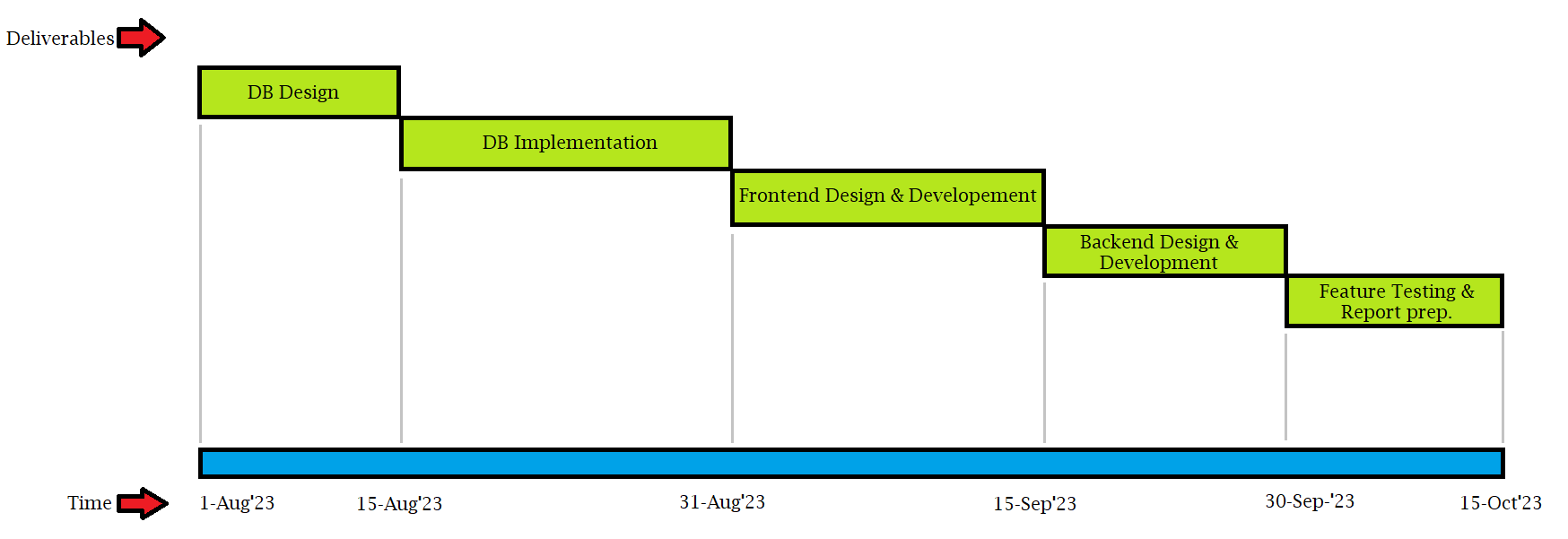
1. **Project features identified**

|  |  |  |
| --- | --- | --- |
| **Feature ID** | **Feature name** | **Description** |
| T01 | Login | Authentication and on success takes you to your dashboard |
|  |  |  |

1. **Software and hardware details**

|  |  |
| --- | --- |
| Platform | Windows |
| Frontend/console | Frontend |
| Backend/server | Backend |
| Database | SQLite |
| Programming Language : Frontend | HTML5, CSS3, Javascript |
| Backend/server: programming Language | Node.js |

1. **Project Plan**



1. **Specify a Google drive link for sharing all your RS documents and future deliverables**
2. **CONCEPTUAL DESIGN PHASE:**

**1. Entity Relationship Model**

Diagram

Description automatically generated

**2.Object Model Diagram (optional)**

Diagram

Description automatically generated

Graphical user interface, application

Description automatically generated

**7. LOGICAL DESIGN PHASE:**

1. **Relational Database Schema**

**From diagram above write down all functional dependencies.**

Eg.,

orderid à custid, productid

1. **Normalization**

Include the below for all the tables.

Table Name: Author

State: 3NF

Reason: No multivalued attributes, No partial dependencies, No transitive dependency

All tables are expected to be in 3NF. Optionally you can choose higher normal forms also. Proper justification for choosing a higher normal form has to be provided.

*Note: Incase if there is any table not in 3NF, it has to be normalized to 3NF and changes made should be projected separately. Changes should also be updated in version table*

1. **CREATE DATA DICTIONARY**

**Table Definitions and Data Contents**

Table

Description automatically generated

1. **PHYSICAL DESIGN PHASE**

NOTE:

Please note the following points w.r.t to deliverable 4 or physical design phase submissions.

1. Each group member should be given a feature or a task in your group.

2. Each member should work on their feature and submit the Relational schema or Model diagram with FDs stated.

3. Since each of you must learn logical design and thereby your individual Relational schema/Relational model (RM) work for the feature allocated will ensure that all of you work and learn.

4. create a folder "DELIVERABLE 4" and in it submit all your individual deliverable4 file which has your allocated feature’s Implemented as SQL code and it should be stored as "<yourname>.sql".

1. SQL Statements

CREATE TABLE [dbo].[Author](

[Fname] [nvarchar](50) NOT NULL, [Minit] [nchar](1) NOT NULL, [Lname] [nvarchar](50) NOT NULL, [DOB] [date] NOT NULL,

CONSTRAINT [PK\_\_Author\_M\_\_EFC884A033D4B598] PRIMARY KEY CLUSTERED

(

[Fname] ASC,

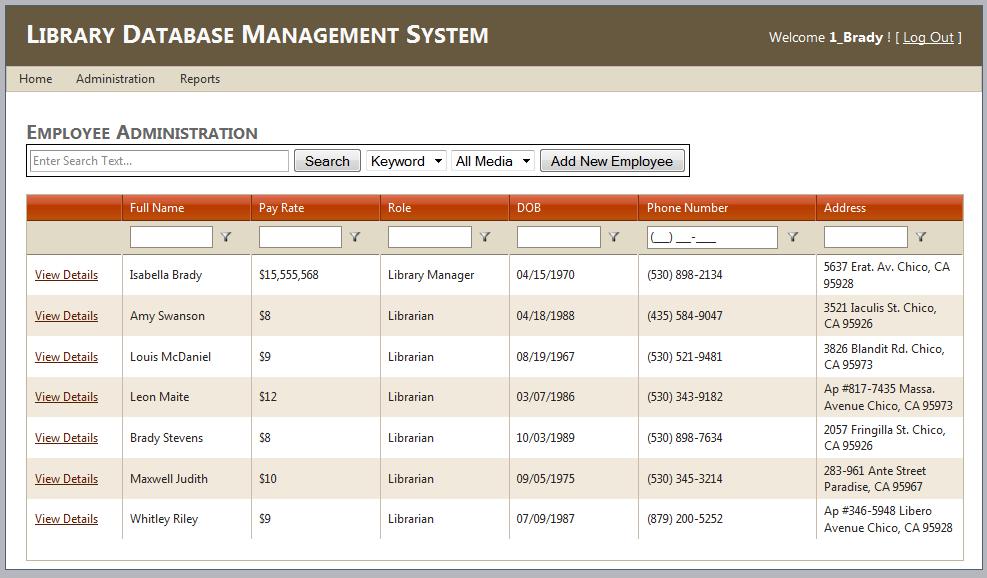
[Lname] ASC, [DOB] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY =OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

1. **Stored Procedures/ Triggers**

Include code for stored procedures/triggers if applicable

1. **User Interface and Database connectivity**
2. Your application should access DB through ODBC or Backend using appropriate technology and demonstrate the DB working.



1. **Using UI themes**

Front end or console based logic or code

**Summary and statistics:**

**For 3 tier systems:**

For Backend:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Feature or functionality | API endpoint name | input | output |
|  |  |  |  |  |

For DB

DB objects used in your project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stored procedure |  |  |  |  |
| Triggers |  |  |  |  |
| JOBS |  |  |  |  |
| Tables |  |  |  |  |
| Views |  |  |  |  |
| Primary Indexes |  |  |  |  |
| Clustered Index |  |  |  |  |
| Secondary Indexes |  |  |  |  |
| Multilevel Indexes |  |  |  |  |
|  |  |  |  |  |

For Frontend

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Feature or functionality | API used | Input passed to API | Output  From API |
|  |  |  |  |  |

**For 1 or 2 tier systems:**

For DB

DB objects used in your project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stored procedure |  |  |  |  |
| Triggers |  |  |  |  |
| JOBS |  |  |  |  |
| Tables |  |  |  |  |
| Views |  |  |  |  |
| Primary Indexes |  |  |  |  |
| Clustered Index |  |  |  |  |
| Secondary Indexes |  |  |  |  |
| Multilevel Indexes |  |  |  |  |
|  |  |  |  |  |

Application End

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Feature or functionality | Function name | Input passed to function | Output  From function |
|  |  |  |  |  |

**12.**

REFERENCES:

1. For Library management System: mattben.info/media/pdf/370\_ImplementationPhase.pdf.

==========================XXXXXX============================