

Challenge #1

Title: Form Approval System

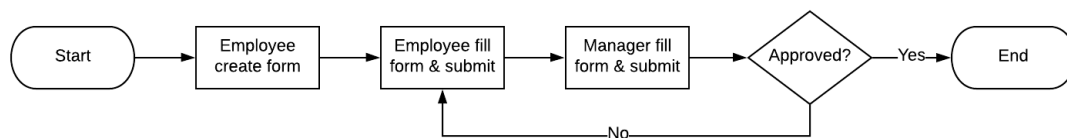
For this challenge, you are required to create a web-based application that utilises the following:

- **Programming languages:** HTML, CSS, JavaScript, PHP
- **Web server:** Apache
- **Databases:** MySQL

Create a Form Approval System which provides form filling and approval functionality. There are three main stakeholders that will be accessing and using the system. They are:

- Admin – maintains all objects in the system
- Manager – approves / rejects forms
- Employee – fills in forms

A typical process flow of the system is as follows:



1. The Employee creates a form, fills in certain sections, and submits the form
2. The Manager edits the form, fills in certain sections, and approves or rejects the form
 - a. If the form is rejected, the Employee needs to re-edit and re-submit the form
3. The process continues until the form is approved

Each of the users can perform the following:

Admin	Manager	Employee
<ul style="list-style-type: none">• View dashboard showing all users and forms• Manage all users• Manage all forms	<ul style="list-style-type: none">• View dashboard showing all forms• Manage forms<ul style="list-style-type: none">○ Approve / reject forms	<ul style="list-style-type: none">• View dashboard showing own forms• Manage own forms

*Manage involves adding, editing, deleting, enabling and disabling the related object

These are some business rules to consider:

- The Employee should not be able to edit the form **after** submission for approval
- The Employee should not be able to edit the form sections that will be filled by the **Manager**
- The Manager should not be able to edit the form **before** submission for approval
- The Manager should not be able to edit the form sections that will be filled by the **Employee**

Example interfaces are shown below:

Index view					
					User
Add					
Owner	Date	Status	Actions		
User A	01/11/2019	Approved	Edit	Delete	
User B	02/11/2019	Pending	Edit	Delete	
User C	03/11/2019	Not approved	Edit	Delete	

This page shows all the forms.

Add view		User
Section A		
Name:	<input type="text"/>	Gender: <input type="radio"/> M <input type="radio"/> F
Address:	<input type="text"/>	Country: <input type="text"/> ▼
Section B		
File 1:	<input type="button" value="Browse"/>	Verified: <input type="checkbox"/>
File 2:	<input type="button" value="Browse"/>	Verified: <input type="checkbox"/>
Section C		
Description:	<input type="text"/>	
Section D		
Reason:	<input type="text"/>	
Approved:	<input type="checkbox"/>	Not Approved: <input type="checkbox"/>
For office use only		
<input type="button" value="Submit"/>		

This page shows an empty form.

Edit view		User
Section A		
Name:	<input type="text" value="User A"/>	Gender: <input checked="" type="radio"/> M <input type="radio"/> F
Address:	<input type="text" value="123, Lorong 11"/>	Country: <input type="text" value="Malaysia"/> ▼
Section B		
File 1:	<input type="text" value="file1.pdf"/>	Verified: <input checked="" type="checkbox"/>
File 2:	<input type="text" value="file2.pdf"/>	Verified: <input checked="" type="checkbox"/>
Section C		
Description:	<input type="text" value="Test"/>	
Section D		
Reason:	<input type="text" value="Test"/>	
Approved:	<input checked="" type="checkbox"/>	Not Approved: <input type="checkbox"/>
For office use only		
<input type="button" value="Submit"/>		

This page shows a form edited by both Employee and Manager.

Challenge #2

Title: School Network Platform

For the first part of this challenge, you are required to create a web-based application that utilises the following:

- **Programming languages:** HTML, CSS, JavaScript, PHP
- **Framework:** Laravel
- **Web server:** Apache
- **Databases:** MySQL

Create a School Network Platform that compiles information from various schools and their students. There are three main stakeholders that will be accessing and using the system. They are:

- Super Admin – maintains all objects in the system
- School Admin – maintains the students under their school
- Student – stores personal and education related information in the system

A typical process flow of the system is as follows:

1. The Super Admin creates a school in the system
2. The Super Admin creates a user account for the School Admin
3. A student registers for an account on the system
4. The School Admin must approve the student before they can login to the system

Each of the users can perform the following:

Super Admin	School Admin	Student
<ul style="list-style-type: none">• View dashboard showing all schools, school admins and students• Manage all schools• Manage all school admins• Manage all students	<ul style="list-style-type: none">• View dashboard showing all students• Approve student registration• Manage students	<ul style="list-style-type: none">• View dashboard showing personal and education related information• Manage personal information• Manage education related information<ul style="list-style-type: none">○ Projects○ Readings

*Manage involves adding, editing, deleting, enabling and disabling the related object

Example interfaces are shown below:

Personal Information Tab

User		
Personal Info	Projects	Readings
<div><div>Name: <input type="text"/></div><div>Gender: <input type="radio"/> M <input type="radio"/> F</div><div>Address: <input type="text"/></div><div>Country: <input type="text"/></div><div>DOB: <input type="text"/></div><div>Phone: <input type="text"/></div><div>Email: <input type="text"/></div><div>....: <input type="text"/></div><div>Submit</div></div>		

This tab will record the personal information of the student.

Projects Tab

User																				
Personal Info		Projects	Readings																	
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="4">Projects</th> </tr> <tr> <th>Title</th> <th>Start Date</th> <th>End Date</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>Project A</td> <td>01/01/2019</td> <td>15/08/2019</td> <td>Completed</td> </tr> <tr> <td>Project B</td> <td>06/06/2019</td> <td>06/06/2020</td> <td>In progress</td> </tr> </tbody> </table>					Projects				Title	Start Date	End Date	Status	Project A	01/01/2019	15/08/2019	Completed	Project B	06/06/2019	06/06/2020	In progress
Projects																				
Title	Start Date	End Date	Status																	
Project A	01/01/2019	15/08/2019	Completed																	
Project B	06/06/2019	06/06/2020	In progress																	
Title: <input style="width: 100%;" type="text"/>																				
Start Date: <input style="width: 150px;" type="text"/>		End Date: <input style="width: 150px;" type="text"/>																		
Status: <input style="width: 100px;" type="text"/> ▼																				
Files: <input style="width: 100px;" type="button" value="Browse"/>																				
Members: <input style="width: 100px;" type="button" value="Find Students"/>																				
<input style="width: 100px;" type="button" value="Submit"/>																				

This tab will record the projects done by the student. The student may link other students (from the same or other schools) who are involved in the project as members.

Readings Tab

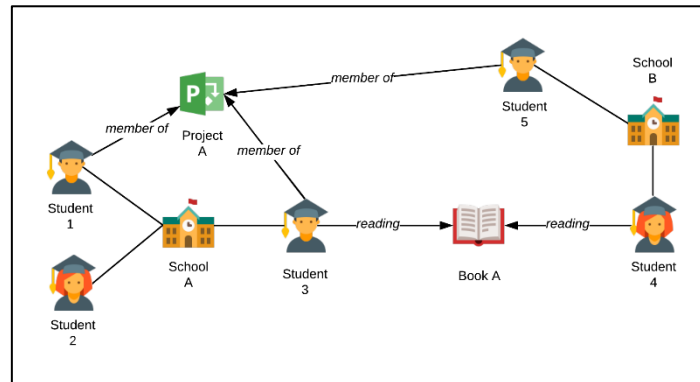
User																				
Personal Info		Projects	Readings																	
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="4">Readings</th> </tr> <tr> <th>Title</th> <th>DOI</th> <th>Year</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>Book A</td> <td>75623</td> <td>2010</td> <td>Book</td> </tr> <tr> <td>Magazine A</td> <td>94383</td> <td>2017</td> <td>Magazine</td> </tr> </tbody> </table>					Readings				Title	DOI	Year	Type	Book A	75623	2010	Book	Magazine A	94383	2017	Magazine
Readings																				
Title	DOI	Year	Type																	
Book A	75623	2010	Book																	
Magazine A	94383	2017	Magazine																	
Title: <input style="width: 100%;" type="text"/>																				
DOI: <input style="width: 150px;" type="text"/>		Year: <input style="width: 150px;" type="text"/>																		
Type: <input style="width: 100px;" type="text"/> ▼																				
<input style="width: 100px;" type="button" value="Submit"/>																				

This tab will record the readings done by a student.

For the second part of this challenge, you are required to create a web-based application that utilises the following:

- **Programming languages:** HTML, CSS, JavaScript, PHP
- **Framework:** Laravel
- **Web server:** Apache
- **Databases:** Graph Database (e.g. Neo4j)

A graph visualisation (shown below) is needed to represent the information stored on the School Network Platform.



The graph visualisation shows all objects in the School Network Platform along with their relationships.

This visualisation should be able to provide:

- An overview of all schools and students
- An overview of a particular school and its students
- An overview of a particular students and their projects / readings
- An understanding of relationships between students
 - Which students are working on the same project?
 - Which students are reading the same materials?
 - Which students (who are not yet project members) can be potential project members of a particular student

Challenge #3

Title: Dynamic Form Versioning System

For this challenge, you are required to create a web-based application that utilises the following:

- **Programming languages:** HTML, CSS, JavaScript, PHP
- **Framework:** Laravel
- **Web server:** Apache
- **Databases:** MySQL / NoSQL

Create a Dynamic Form Versioning System which allows users to create, fill, and submit forms that are based on different template versions. For example, assume there are three versions of a form as shown below:

Form Version 1	Form Version 2	Form Version 3
<ul style="list-style-type: none">• Name• Date of birth• Gender• Address	<ul style="list-style-type: none">• Name• Date of birth• Gender• Address• Phone number• Email	<ul style="list-style-type: none">• Name• Location<ul style="list-style-type: none">○ Address○ Phone number○ Email

A typical process flow of the system is as follows:

1. A user creates a form by selecting from a list of template versions
2. Depending on the version selected, the form shown to the user would contain specific fields
 - a. E.g. form version 1 would have name, date of birth, gender and address
 - b. E.g. form version 2 has additional fields such as phone number and email
 - c. E.g. form version 3 has removed the date of birth and gender fields. However, in this version, multiple locations can be added, each of which has an address, phone number and email
3. After submitting the form, the information shall be stored into the database

These are some requirements to consider:

- The database should be able to accommodate different form versions without having to constantly change its structure. Therefore, a NoSQL database approach is suggested
 - This should **not** be done: whenever there is a new version of the form, create another database table
- The database structure should allow queries to obtain statistical information from all form versions. For instance:
 - The count of all male / female in all Form Versions
 - The count of all male in Form Version 1
 - A list of email addresses in Forms Version 2 and 3
 - The count of people aged between 21 – 30 years old in all Form Versions

Example interfaces are shown below:

Index view				
User				
Add				
Owner	Date	Version	Actions	
User A	01/11/2019	1	Edit	Delete
User B	02/11/2019	2	Edit	Delete
User C	03/11/2019	3	Edit	Delete

This page shows all the forms.

Add view	
User	
Version:	<div><div>1</div><div>2</div><div>3</div><div>4</div><div>5</div></div>
<div>Create</div>	

This page shows the selection of form version.

Edit view	
User	
Version 2	
Name:	<div>User A</div>
DOB:	<div>01/01/90</div>
Address:	<div>Home address</div>
Phone:	<div>1234</div>
Gender:	<div><input type="radio"/> M <input type="radio"/> F</div>
Email:	<div>email@home.com</div>
<div>Submit</div>	

This page shows editing Form Version 2.

Edit view	
User	
Version 3	
Name:	<div>User A</div>
Location 1	
Address:	<div>Home address</div>
Phone:	<div>1234</div>
Email:	<div>email@home.com</div>
Location 2	
Address:	<div>Work address</div>
Phone:	<div>6789</div>
Email:	<div>email@work.com</div>
<div>+ Location</div>	
<div>Submit</div>	

This page shows editing of Form Version 3.