# Project Guidelines

Hello Students,

This is your group project, the most integral part of this course. It is not just an assignment but something you will discuss in future job interviews. Expect questions such as:

- How did you build your framework?
- How did you collaborate within your team?
- What classes did you implement in your framework?
- How did you apply the Page Object Model (POM)?
- How did you approach UI, API, and Database testing?

This project will serve as your foundation to confidently answer these questions, so put in your best effort!

**Key Focus Area: Writing Effective Test Cases**

One of the most critical aspects of this project is writing comprehensive test cases based on user stories. Ensure that your test cases fully cover the acceptance criteria of each story.

When writing feature files, focus on reusability and scalability by using:
✅ Parameters
✅ Scenario Outlines
✅ Data Tables (where applicable)

Make sure you implement the most efficient approach to enhance the flexibility of your framework.

## Framework Preparation

Each team must set up the base structure of the framework, ensuring the following components are ready:

- Project dependencies configured
- Configuration file set up
- Utility classes built from scratch (no copy-pasting)
- Core pages (e.g., Login, Dashboard) implemented
- One login feature fully developed
- Step Definitions and Runner Class included

Collaboration: Every team member must actively participate in building the framework. Once each person completes their assigned part, then the code review needs to be completed with the team and the team lead will integrate the code to ensure a complete and functional framework.

Understanding the Framework: Each team member should understand the framework's structure and how different components interact.

---

## Sprint Grooming

- Review all user stories with the team.
- Ensure every team member understands the requirements.
- Assign story points to each user story based on complexity.
- Sprint duration: 1 week

---

## Sprint Planning

- Each team member selects at least one user story they are comfortable working on.
- Break down the user story into smaller tasks and assign responsibilities accordingly.

---

## Development & Testing Process

Phase 1: Manual Testing

- Perform manual testing on assigned features before automating them.

Phase 2: Automation Development

- Start automating the user stories after verifying functionality through manual testing.

---

## Team Collaboration & Support

- If a team member faces any difficulties during the project or automation process, the team must work together to resolve issues.
- Communication and teamwork are essential for project success.

---

## Final Steps (Code Completion & Best Practices)

Before finalizing the project, ensure the following:

✅ Methods for handling radio buttons, checkboxes, dropdowns, alerts, frames, and windows are implemented.
✅ The framework executes smoothly using Maven and run through Jenkins
✅ Automated tests generate reports with screenshots.
✅ All user stories are fully automated and code is stored in GitHub.

**Git Workflow:**

To keep the process simple, follow this approach:

1. Cloning the Repository: Each student clones the project repo to get the base framework.
2. Working on Assigned Parts: Students develop their assigned sections on their local machines.

3. Sharing Code with the Team Lead: Once completed, students send their code to the team lead via email/chat and discuss it in a meeting.
4. Team Lead Pushes Code: The team lead integrates all contributions and pushes updates to GitHub.
5. Pulling Updates from GitHub: Students pull the latest changes before continuing their work.

---

## 7. Jenkins Integration

After completing the framework, each team must:

- Create a Maven project job on Jenkins.
- Run the automation project through Jenkins.
- Add Cucumber Maven Reports to the Jenkins build process.
- Copy the Jenkins Maven report link for final submission.

---

## 8. Sprint Retrospective & Final Submission

- Each team should draw a diagram of their framework to showcase its structure.
- Record a group video explaining the framework and project process.

---

**Each member must pick at least 1 user story to automate**

## USER STORY 1: Admin User Login for HRMS Application

As an HRMS administrator,
I want to be able to log in to the HRMS application securely with my credentials,
So that I can access the admin dashboard.

**Acceptance Criteria:**

1. An admin user with valid login credentials should be able to successfully access the HRMS application.
2. The login process should involve entering a valid username and password combination.
3. Upon successful login, the admin user should be redirected to the appropriate admin dashboard or landing page within the HRMS application.

Application URL:
http://hrm.syntaxtechs.net/humanresources/symfony/web/index.php/auth/login

username: Admin
password: Hum@nhrm123

## USER STORY 2: Adding an Employee to the HRMS Application

As an HRMS application administrator,
I want to be able to add new employees to the HRMS system with or without specifying an employee ID,
So that the system can manage employee records effectively, assigning unique IDs as needed.

**Acceptance Criteria:**

1. An admin user should be able to add an employee to the HRMS application without providing an employee ID. The system should automatically generate a unique employee ID for the new employeeThe required fields for this process should include:
   - First Name

  - Middle Name (optional)
  - Last Name
2. An admin user should also have the option to add an employee to the HRMS application by providing a unique employee ID. The required fields for this process should include:
   - First Name
   - Middle Name (optional)
   - Last Name
   - Employee ID
3. The system should provide appropriate error messages or prompts for users who attempt to submit incomplete or invalid employee information.
4. Verify the employee record from the database as well.

   HRMS url:
   http://hrmstest.syntaxtechs.net/humanresources/symfony/web/index.php/auth/login

   **Admin user details:**

   username: Admin
   password: Hum@nhrm123

## USER STORY 3: Creating Login details for an Employee in the the HRMS Application

As an HRMS application administrator,
I want to create login credentials for a new employee in the HRMS system,
So that the employee can access the HRMS application with unique login details.

**Acceptance Criteria:**

1. An admin user should be able to create login details for an employee to the HRMS application . The system should automatically generate a unique employee ID for the new employee. The required fields for this process should include:
   - First Name: There must be a mandatory textbox for entering the first name
   - Middle Name: There must be an optional textbox for entering the middle name
   - Last Name: There must be a mandatory textbox for entering the last name.
   - Create Login Details: A checkbox titled "Create Login Details" must be present. The username and password fields should only be enabled if this box is checked.

- User Name: A mandatory field should be provided for entering a unique username.
- Password: A mandatory field should be provided for entering a password, which must meet predefined complexity requirements.

  (Hint Must display following text: For a strong password, please use a hard to guess combination of text with upper and lower case characters, symbols and numbers).

- Confirm Password: A mandatory field where the user must re-enter the password for verification. The system should validate that this matches the 'Password' field before allowing the process to continue.
- Status: A dropdown for 'Status' must be provided with predefined status options: Enabled, Disabled

2. Upon successful submission of the login details, the system should save the changes and employee with created credential must be able to login to the HRMS application
   Note: please make sure to save these details to use for future testing purposes

## USER STORY 4: Search for an Employee in the HRMS Application

As an HRMS application administrator,
I want to search for employees by their name or employee ID,
So that I can quickly locate and access their records in the HRMS system.

**Acceptance Criteria:**

1. As an admin user, I should be able to search for an employee using their full name (first and last name) as the search criteria. The search should return all matching employee records.
2. The search functionality should be flexible and capable of handling partial name matches and variations in capitalization.
3. As an admin user, I should be able to search for an employee using their unique employee ID as the search criteria. The search should return the exact employee record associated with the provided employee ID.
4. The system should provide a message “No Records Found” if no matching employee records are found

HRMS url:
http://hrmstest.syntaxtechs.net/humanresources/symfony/web/index.php/auth/login

username: Admin
password: Hum@nhrm123

## USER STORY 5: As an ESS user, I should be able to change my personal details

As an ESS user,
I want to be able to edit my personal information within the HRMS application,
So that I can keep my details accurate and up-to-date.

**Acceptance Criteria:**

1. The employee should be able to access and edit their personal information, including the following fields:
   - First Name: A textbox should be displayed where the employee can enter or update their first name.
   - Middle Name: A textbox should be displayed where the employee can enter or update their middle name (optional).
   - Last Name: A textbox should be displayed where the employee can enter or update their last name.
   - Gender: A radio button should be displayed where the employee can select or update their gender (Male, Female).
   - Nationality: A dropdown should be displayed where the employee can select or update their nationality.
   - Marital Status: A dropdown should be displayed where the employee can select or update their marital status (Single, Married, Other).

2. Upon successful submission of the updated personal information, the system should save the changes.
3.Verify also the changes are made in the database as well.

HRMS url:
http://hrmstest.syntaxtechs.net/humanresources/symfony/web/index.php/auth/login

Note: please use ESS user details from the story Creating Login details for an Employee in the the HRMS Application

## USER STORY 6: As an employee, I should be able to change my contact details.

As an employee (ESS user),
I want to be able to edit my contact information within the HRMS application,
So that my contact details remain current and accurate.

**Acceptance Criteria:**

1. The employee should be able to access and edit their contact information, including the following fields:
    - Address Street 1: A textbox should be displayed where the employee can enter or update their address.
    - Address Street 1: A textbox should be displayed where the employee can enter or update their address.
    - City: A textbox should be displayed where the employee can enter or update their city.
    - State: A textbox should be displayed where the employee can enter or update their state.
    - Zip Code: A textbox should be displayed where the employee can enter or update their zip code
    - Country: A dropdown should be displayed where the employee can select or update their country.
    - Home Phone: A textbox should be displayed where the employee can enter or update their home phone number
    - Mobile Phone: A textbox should be displayed where the employee can enter or update their mobile phone number
    - Work Phone: A textbox should be displayed where the employee can enter or update their work phone number
    - Work Email: A textbox should be displayed where the employee can enter or update their work email address
    - Other email: A textbox should be displayed where the employee can enter or update their personal or other email address

2. Upon successful submission of the updated contact information, the system should save the changes.

HRMS url:
http://hrmstest.syntaxtechs.net/humanresources/symfony/web/index.php/auth/login

Note: please use ESS user details from the story Creating Login details for an Employee in the the HRMS Application

## USER STORY 7: Profile Picture Upload for ESS Users

As an ESS user,
I want to upload a profile picture to personalize my profile on the HRMS platform,
So that my profile feels more complete and personalized.

**Acceptance Criteria:**

As an ESS user, I should be able to upload a profile picture to enhance my personalization on the platform.

Requirements:

- The user should be able to upload a profile picture within the specified guidelines.
- Supported file formats for the profile picture should include .jpg, .png, and .gif.
- The maximum file size allowed for the profile picture should be 1MB.
- Recommended dimensions for the profile picture should be 200px x 200px for optimal display and user experience.
- The platform should provide appropriate error messages or prompts for users who attempt to upload images that do not meet the specified criteria.
- Upon successful upload, the user's profile should display the updated profile picture.

  HRMS url:
  http://hrmstest.syntaxtechs.net/humanresources/symfony/web/index.php/auth/login

Note: please use ESS user details from the story Creating Login details for an Employee in the the HRMS Application

## USER STORY 8: As an employee, I should be able to add dependents to my profile in the HRMS application.

As an employee or admin,
I want to be able to add dependents to an employee's profile within the HRMS application,
So that dependent information is accurately recorded and easily accessible.

**Acceptance Criteria:**

1. The employee should be able to access a dedicated section within the HRMS application to add dependents.
2. The following fields must be displayed and editable for adding a dependent:

- **Name:** A textbox should be displayed where the employee can enter the full name of the dependent.
- **Relationship:** A dropdown should be displayed where the employee can select the relationship of the dependent (e.g., Spouse, Child, Parent, Sibling, Other).
- **Date of Birth:** A calendar or date input field should be displayed where the employee can select or enter the dependent's date of birth.

3. The employee should be able to add multiple dependents, and the HRMS application should clearly display the list of dependents added by the employee.
4. The system should provide appropriate error messages or prompts for users who submit incomplete or invalid dependent information. Error messages should be clear, easily visible, and placed near the respective input fields.
5. The employee should be able to review, edit, or remove dependents as needed.
6. Also verify the changes through database.

## USER STORY 9: As an admin, I want to be able to add current job details for employees.

As an HRMS admin user,
I want to be able to add and update current job details for employees in the HRMS application,
So that employee records are complete and reflect accurate job information.

**Acceptance Criteria:**

1. The Admin user can access the employee's profile and navigate to the current job details section.
   The following fields should be displayed and editable for the Admin user:
   - Job Title: a dropdown field where the Admin user can select the job title of the employee.
   - Employment Status: a dropdown field where the Admin user can select the employment status of the employee (Full-time, Part-time, Contract, etc.).
   - Job Category: a dropdown field where the Admin user can select the job category of the employee (Sales, Marketing, HR, etc.).
   - Joined Date: a calendar where the Admin user can select the date when the employee joined the organization.
   - Sub Unit: a dropdown field where the Admin user can select the sub unit where the employee is currently working.

- Location: a dropdown field where the Admin user can select the location where the employee is currently working.
- Start Date: a calendar where the Admin user can select the start date of the employee's current job.
- End Date: a calendar where the Admin user can select the end date of the employee's current job (if applicable).
- Contract Details: a file upload field that accepts files up to 1MB where the Admin user can upload the employee's contract details (if applicable).
- The system should validate that mandatory fields (Job Title, Employment Status, Joined Date, Sub Unit, Location) are filled in before allowing the Admin user to save the current job details.
- After the Admin user saves the current job details, the system should store the current job details in the employee's profile.
- Also verify the changes are reflected in the database.

## USER STORY 10: As an employee, I want to be able to add my language proficiency details.

As an employee,
I want to be able to add my language proficiency details to my HRMS profile,
So that my profile accurately reflects my language skills.

**Acceptance Criteria**

1. The employee can access their profile and navigate to the language proficiency section.
    - The following fields should be displayed and editable for the employee:
    - Language: a dropdown field where the employee can select the language they are proficient in.
    - Fluency: a dropdown field where the employee can select their level of fluency in the language (Basic, Intermediate, Advanced, Fluent).
    - Competency: a dropdown field where the employee can select their competency level in the language (Speaking, Reading, Writing).
    - Comments: a text box where the employee can enter additional comments or details about their language proficiency.
    - The system should validate that mandatory fields (Language, Fluency, Competency) are filled in before allowing the employee to save their language proficiency details.

    - After the employee saves their language proficiency details, the system should store the language proficiency details in the employee's profile.
2. The employee should be able to edit or delete their previously saved language proficiency details.

## USER STORY 11: As an employee, I want to be able to add my membership details.

As an employee,
I want to be able to add my membership details to my HRMS profile,
So that my profile accurately reflects my professional memberships and subscriptions.

**Acceptance Criteria**

1. The employee can access their profile and navigate to the membership section.
   The following fields should be displayed and editable for the employee:
    - Membership: a dropdown field where the employee can select the membership type.
    - Subscription Paid By: a dropdown field where the employee can select the entity that paid for the subscription (Employee, Employer, Other).
    - Subscription Amount: a text box where the employee can enter the subscription amount.
    - Currency: a dropdown field where the employee can select the currency used for the subscription payment.
    - Subscription Commence Date: a calendar where the employee can select the date when the subscription started.
    - Subscription Renewal Date: a calendar where the employee can select the date when the subscription will renew.
    - The system should validate that mandatory fields (Membership, Subscription Paid By, Subscription Amount, Subscription Commence Date) are filled in before allowing the employee to save their membership details.
    - After the employee saves their membership details, the system should store the membership details in the employee's profile.
2. The employee should be able to edit or delete their previously saved membership details.
    - The membership details should be displayed in a clear and organized format on the employee's profile page.

3. Also verify the reflected changes in database.

## API User Story 1: User Creation

As a tester,I want to validate that the API for creating a new user account functions correctly,So that users can register successfully, and appropriate error messages are displayed when inputs are invalid

**Acceptance Criteria:**

- The API should return a success message "User Created" with the user's details when all required fields (name, email, password) are provided correctly and the email is unique.
- If a user attempts to register with an email that is already in the system, the API should respond with the message "The email address you have entered is already registered" to indicate that duplicate emails are not allowed.
- The API should return "condition": "error" and "data": "Invalid Email" if the provided email format is incorrect (e.g., missing the "@" symbol).
- If the password field is missing or empty, the API should respond with "condition": "error" and "data": "Please fill all inputs" to ensure all fields are completed.
- If the name field is missing or empty, the API should also respond with "condition": "error" and "data": "Please fill all inputs" to prompt the user to fill all required fields.

## API User Story2: Generate Authentication Token

As a tester,I want to validate that the API for generating an authentication token functions correctly,So that a user can receive a token to perform subsequent actions, such as creating or modifying employee records

Acceptance Criteria:

- The API should return a token in the response when valid email and password credentials are provided. The token should follow a JWT format and be valid for authentication in further requests.
- If the email format is incorrect or if the email/password combination does not match an existing user, the API should respond with an error message: "Error": "Email or Password is incorrect."
- If any required fields, such as the password, are missing, the API should return an error message: "Error": "Please fill all inputs" to indicate that all fields must be completed.

## **API User Story-3:** Create Employee Record

As a tester,I want to validate the API for creating a new employee record,So that employee details are stored correctly, and error handling works for invalid or incomplete data

**Acceptance Criteria:**

- When all required fields are provided with valid data, the API should respond with "Message": "Employee Created" and return the employee details, including a unique "employee_id".
- If the gender field contains an invalid value, such as anything other than "M" or "F", the API should return an error message: "Message": "enter M for male enter F for female".
- If the birthday field is provided in an incorrect date format (e.g., not in "yyyy-mm-dd" format), the API should return an error message: "Error": "Please Enter a Valid Date Format. Example \"yyyy-mm-dd\"".
- If any required fields are missing or empty (e.g., emp_status), the API should return an error message indicating that the payload is incomplete, such as:
    i. "Error": "Enter Full Payload"
    ii. "Message": "Status is Empty" if the emp_status field is specifically missing.

## **API User Story 4:** Retrieve Employee Record

As a tester,I want to validate the API for retrieving an existing employee record using the employee ID,So that users can access accurate details of a specific employee

**Acceptance Criteria:**

1. When a valid employee ID is provided, the API should return the employee's complete details, including employee_id, emp_firstname, emp_middle_name, emp_lastname, emp_birthday, emp_gender, emp_job_title, and emp_status.
2. If the provided employee ID does not exist in the system or is invalid, the API should return an error message:Employee does not exist or you have provided invalid employee_id

## API User Story 5: Delete Employee Record

As a tester,I want to validate the API for deleting an existing employee record using the employee ID,So that the employee's information is removed from the system, and appropriate error messages are displayed when accessing deleted records

**Acceptance Criteria:**

1. The API should return a success message "Employee deleted" along with the deleted employee's details (e.g., employee_id, emp_firstname, emp_middle_name, emp_lastname, emp_birthday, emp_gender) when a valid employee_id is provided for deletion.
2. If a request is made to delete or retrieve an employee with an ID that does not exist in the system (such as after it has been deleted), the API should respond with the message "Employee does not exist or you have provided invalid employee_id".
3. If the employee_id parameter is missing from the delete request, the API should return an error message indicating that employee_id is required for deletion.

   Error :Please provide employee_id