Name ……

v1.0.0

STUDY MATERIALS

- Coursera: https://www.coursera.org/

  username : k.manandhar@plieger.nl

Internship Timeline

1. HTML,CSS & JS ……..…….3 weeks
2. PHP ……..……...……...….2 weeks
3. VUEJS 3 ….……..……...……..2 weeks
4. LARAVEL…..…….……..…….…….……2 weeks
5. POSTMAN……...……..3 days
6. GIT……..……...2 days
7. VUEJS 3 & LARAVEL…..……..……..…...….6 weeks
8. REAL TIME PROJECT…….……...…..4 weeks

Note: Mock test shall be conducted in following order:

Moct Test 1: After completion of HTML,CSS & JS (1) and PHP (2)
Full Marks: 20
Time: 30 minutes

Moct Test 2: After completion of VUEJS 3 (3) and LARAVEL(4)
Full Marks: 30
Time: 45 minutes

Moct Test 3: After completion of POSTMAN (5) and GIT (6)
Full Marks: 8
Time: 15 minutes

Moct Test 4: After completion of VUEJS 3 & LARAVEL (7)
Full Marks: 52
Time: 90 minutes

Pass marks shall be calculated as per percentile basis.

Note: Marks of each mock test shall be recorded and accumulated for final evaluation at the end of the Internship.

To attend soft skills training and for real time project, you must pass the exam

…….
Date of Start

Best of Luck !

HTML,CSS & JS

1st Week

Day 1: Introduction to HTML

- What is HTML and why it's important
- HTML syntax and structure
- Basic HTML tags: headings, paragraphs, links, images

Task**:** Create a simple HTML page with headings, paragraphs, links, and images.

Reference links:

- HTML Introduction
- HTML Basic Tags

Day 2: HTML Tables and Forms

- Creating HTML tables for displaying data
- HTML form elements for user input
- Common form elements: text input, radio buttons, checkboxes, dropdowns

Task**:** Create an HTML table to display data and an HTML form to collect user input.

Reference links:

- HTML Tables
- HTML Forms

Day 3: Introduction to CSS

- What is CSS and why it's important
- CSS syntax and selectors
- Basic CSS properties: color, background, font, text

Task**:** Add CSS styles to an HTML page to change the color, background, font, and text properties.

Reference links:

- CSS Introduction
- CSS Selectors

Day 4: CSS Layouts

- CSS box model: margin, padding, border
- CSS layout: positioning, float, display
- Responsive design: media queries

Task**:** Create a simple CSS layout using positioning, float, and display properties, and add media queries for responsiveness.

Reference links:

- CSS Box Model
- CSS Layout
- CSS Media Queries

Day 5: Introduction to JavaScript

- What is JavaScript and why it's important
- JavaScript syntax and data types
- Basic JavaScript functions and operators

Task**:** Write a simple JavaScript function that performs a basic operation on data types.

Reference links:

- JavaScript Introduction
- JavaScript Data Types
- JavaScript Functions

Day 6: Revision

2nd Week

Day 7: JavaScript Conditionals and Loops

- JavaScript conditional statements: if, else, switch
- JavaScript loops: for, while, do-while
- Common JavaScript array methods: push, pop, shift, unshift

Task**:** Write a JavaScript program that uses conditional statements and loops to manipulate data in an array.

Reference links:

- JavaScript Conditional Statements
- JavaScript Loops
- JavaScript Arrays

Day 8: DOM Manipulation with JavaScript

- What is the Document Object Model (DOM) and why it's important
- JavaScript DOM methods: getElementById, getElementsByTagName, createElement
- Manipulating HTML content and styles with JavaScript

Task**:** Write a JavaScript program that manipulates HTML content and styles using the DOM methods.

Reference links:

- JavaScript DOM

Day 9: JavaScript Events

- What are events and how they work
- JavaScript event listeners: addEventListener, removeEventListener
- Common HTML events: click, mouseover, keypress

Task**:** Write a JavaScript program that uses event listeners to respond to user actions on an HTML page.

Reference links:

- JavaScript Events
- JavaScript Event Listeners

Day 10: CSS Flexbox

- What is CSS Flexbox and why it's important
- Flexbox properties: display, flex-direction, justify-content, align-items
- Creating responsive layouts with Flexbox

Task**:** Create a responsive layout using CSS Flexbox.

Reference links:

- CSS Flexbox
- CSS Flexbox Froggy

Day 11: CSS Grid

- What is CSS Grid and why it's important
- Grid properties: display, grid-template-columns, grid-template-rows, grid-gap
- Creating complex layouts with CSS Grid

Task**:** Create a complex layout using CSS Grid.

Reference links:

- CSS Grid
- CSS Grid Garden

Day 12: Revision

3rd Week

Day 13-16: Mini project

- Apply HTML, CSS, and JavaScript knowledge to create a mini project
- Project requirements and specifications will be provided to the interns
- Mentors will provide guidance and feedback during the project development

Task**:** Develop a mini project applying the HTML, CSS, and JavaScript knowledge gained during the internship.

Reference links:

- HTML, CSS, and JavaScript projects ideas
- Bootstrap (optional)

Day 17-18: Debugging and testing

- Debugging techniques: console.log, breakpoints, step through code
- Testing methodologies: unit testing, integration testing, end-to-end testing

Task**:** Debug and test the mini project developed during the previous days.

Reference links:

- JavaScript Debugging
- Testing JavaScript code
- Jest (optional)

PHP

1st Week

Day 1: Introduction to PHP

- What is PHP and its role in web development
- Basic PHP syntax: variables, data types, operators
- Writing and running PHP code

Task**:** Write a simple PHP program to print out "Hello, World!" on a web page.

Reference links:

- PHP Introduction
- PHP Syntax

Day 2: Control Structures

- If-else statements
- Switch statements
- Loops: for, while, do-while, foreach, forelse, map, each

Task**:** Write a PHP program to determine if a number is odd or even using if-else statements.

Reference links:

- PHP Control Structures

Day 3: Arrays and Functions

- Creating and manipulating arrays in PHP
- Creating and calling functions in PHP
- Passing and returning arguments to functions

Task**:** Write a PHP function to calculate the sum of an array of numbers.

Reference links:

- PHP Arrays
- PHP Functions

Day 4: Forms and Superglobals

- Creating HTML forms to collect user input
- Accessing form data with PHP
- Understanding superglobals: $_GET, $_POST, $_REQUEST

Task**:** Create an HTML form to collect a user's name and use PHP to display a personalized greeting.

Reference links:

- PHP Forms
- PHP Superglobals

Day 5: File Handling

- Reading and writing files with PHP
- Understanding file permissions
- Uploading files with PHP

Task**:** Write a PHP program to read a file and display its contents on a web page.

Reference links:

- PHP File Handling

Day 6: Revision

2nd Week

Day 7-8: MySQL Database

- Introduction to MySQL and its role in web development
- Creating a MySQL database and tables
- Connecting to MySQL using PHP
- Performing CRUD operations: creating, reading, updating, and deleting data in the database

Task**:** Create a MySQL database to store user information and use PHP to add, retrieve, and update data in the database.

Reference links:

- MySQL Tutorial
- PHP MySQL

Day 9: Object-Oriented Programming

- Understanding the principles of object-oriented programming (OOP)
- Creating classes, traits and objects in PHP
- Using inheritance and polymorphism in PHP

Task**:** Write a PHP program using OOP to create a calculator class with basic arithmetic operations.

Reference links:

- PHP OOP

Day 10: Exception Handling

- Understanding exceptions and how they occur in PHP
- Catching and handling exceptions
- Using try-catch blocks in PHP

Task**:** Write a PHP program to handle an exception that occurs when dividing a number by zero.

Reference links:

- PHP Exceptions

Day 11: PHP Frameworks

- Introduction to PHP frameworks and their benefits
- Popular PHP frameworks: Laravel, CodeIgniter, Symfony
- Using a PHP framework to build web applications

Task**:** Choose a PHP framework (e.g. Laravel) and build a simple web application that connects to a database and displays data on a web page.

Reference links:

- Best PHP Frameworks
- Laravel Tutorial
- CodeIgniter Tutorial
- Symfony Tutorial

This syllabus should give you a good foundation in PHP programming. Good luck with your studies!

Day 12: Revision

VUEJS 3

1st Week

Day 1: Introduction to Vue.js 3

- What's new in Vue.js 3
- Setting up a Vue.js 3 project
- The Composition API

Task**:** Create a simple Vue.js 3 app that displays a message on a web page.

Reference links:

- Vue.js 3 Introduction
- Vue.js 3 Installation
- Vue.js 3 Composition API

Day 2: Directives and Data Binding

- Understanding directives and their role in Vue.js
- Using directive to manipulate the DOM
- Understanding data binding in Vue.js 3

Task**:** Create a Vue.js 3 app that uses directives and data binding to display dynamic content on a web page.

Reference links:

- Vue.js 3 Directives
- Vue.js 3 Data Binding

Day 3: Components

- Creating and using components in Vue.js 3
- Understanding component lifecycle hooks
- Communicating between components

Task**:** Create a Vue.js 3 app that uses components to display data in a modular and reusable way.

Reference links:

- Vue.js 3 Components
- Vue.js 3 Component Lifecycle

Day 4: Routing

- Creating and using routes in Vue.js 3
- Understanding dynamic routes
- Navigating between routes

Task**:** Create a Vue.js 3 app that uses routing to navigate between different pages.

Reference links:

- Vue.js 3 Routing
- Vue.js 3 Dynamic Routing

Day 5: Vuex

- Understanding the role of Vuex in Vue.js 3
- Creating and using Vuex stores
- Using Vuex for state management

Task**:** Create a Vue.js 3 app that uses Vuex for state management.

Reference links:

- Vue.js 3 Vuex

Day 6: Revision

2nd Week

Day 7-12: Vue.js 3 Project

- Using Vue.js 3 to build a simple crud web application
- Combining the concepts learned in the previous days to create a fully functional Vue.js 3 project

Task**:** Create a Vue.js 3 project that includes components, routing, and Vuex.

Reference links:

- Vue.js 3 Projects
- Vue.js 3 Examples

This syllabus should give you a good foundation in Vue.js 3 programming. Good luck with your studies!

LARAVEL

1st Week

Day 1: Introduction to Laravel

- Understanding the Laravel framework
- Setting up a local development environment with Laravel
- Creating a new Laravel project

Task**:** Set up a local development environment with Laravel and create a new Laravel project.

Reference links:

- Laravel Documentation
- Laravel Installation Guide

Day 2: Routing and Controllers

- Understanding Laravel's routing with named, grouped, prefix, parameters, invalid,route controller, resource and resources route,
- Implementing CRUD operations using controllers

Task**:** Create a Laravel app with multiple routes and controllers, and implement CRUD operations.

Reference links:

- Laravel Routing
- Laravel Controllers

Day 3: Views and Blade Templating

- Understanding Laravel's Blade templating engine
- Creating and using views in Laravel
- Implementing layouts, partials, component with Blade
- Passing data to component and rendering the component with data

Task**:** Use Blade to create views with layouts and partials in a Laravel app.

Reference links:

- Laravel Views
- Laravel Blade Templating

Day 4: Models and Eloquent ORM

- Understanding Laravel's Eloquent ORM
- Creating and using models in Laravel
- Implementing CRUD operations using Eloquent

Task**:** Use Eloquent to create and manage models and implement CRUD operations.

Reference links:

- Laravel Models
- Laravel Eloquent ORM

Day 5: Database Migrations,Seeders and factories

- Understanding Laravel's database migrations
- Creating and running migrations in Laravel
- Using seeders to populate a database with test data
- Using factories to populate a database with test data

Task**:** Use migrations to create and modify database tables, and use seeders to populate the database with test data.

Reference links:

- Laravel Migrations
- Laravel Seeders
- Laravel Factories

Day 6: Revision

2nd Week

Day 7-12: Laravel Project

- Using Laravel to build a simple crud web application
- Combining the concepts learned in the previous days to create a fully functional Laravel project

Task**:** Create a Laravel project that includes routing, controllers, views, models, Eloquent, migrations, and seeders.

Reference links:

- Laravel Projects

This syllabus should give you a good foundation in Laravel programming. Good luck with your studies!

POSTMAN

Day 1: Introduction to Postman and Sending Requests

- Introduction to Postman: Learn about Postman, its features, and its uses.
    - Reference link: What is Postman?
- Installation and Setup: Install and set up Postman on your computer.
    - Reference link: Installing Postman
- Sending GET Requests: Learn how to send GET requests in Postman.
    - Reference link: Sending a GET request
- Response Types: Understand the different types of responses you can receive from an API.
    - Reference link: Response Types

Tasks:

- Install Postman on your computer.
- Send a GET request to the Postman Echo API.
- Send a GET request to the OpenWeatherMap API to get the current weather for a city of your choice.

Day 2: Request Body, Authentication, and Testing APIs

- Request Body and Parameters: Learn how to send requests with request bodies and parameters.
    - Reference link: Request Body
- Authentication and Headers: Understand how to authenticate your requests and use headers in Postman.
    - Reference link: Authentication
- Testing APIs: Learn how to write tests for your APIs in Postman.
    - Reference link: Writing Tests

Tasks:

- Send a POST request to the Postman Echo API with a request body.
- Send a request to the Github API with authentication.
- Write tests for your requests in Postman.

Day 3: Variables, Newman, and Mocking APIs

- Variables and Data Files: Learn how to use variables and data files in Postman.
    - Reference link: Variables
- Newman Command Line Tool: Understand how to use Newman to run your Postman collections from the command line.
    - Reference link: Newman
- Mocking APIs: Learn how to mock APIs in Postman.
    - Reference link: Mock Servers

Tasks:

- Use variables and data files in your requests in Postman.
- Use Newman to run a Postman collection from the command line.
- Create a mock API in Postman.

GIT

Day 1: Git Basics

- Introduction to Git: Learn about Git, its features, and its uses.
    - Reference link: What is Git?
- Installation and Setup: Install and set up Git on your computer.
    - Reference link: Installing Git
- Creating a Repository: Learn how to create a repository in Git.
    - Reference link: Creating a Repository
- Making Changes: Understand how to make changes to your repository and commit them in Git.
    - Reference link: Making Changes

Tasks:

- Install Git on your computer.
- Create a Git repository on your computer and add some files to it.
- Commit changes to your Git repository.

Day 2: Branching, Merging, and Collaborating

- Branching and Merging: Learn how to create and switch between branches in Git and how to merge changes from one branch to another.
    - Reference link: Branches, Merging

- Collaborating with Others: Understand how to collaborate with others using Git, including pushing and pulling changes and resolving merge conflicts.
    - Reference link: Collaborating

Tasks:

- Create a new branch in your Git repository, make some changes, and merge them back into the main branch.
- Collaborate with someone else by setting up a remote repository, pushing and pulling changes, and resolving merge conflicts.

VUEJS 3 & LARAVEL

1st Week

Day 1: Introduction to HTML, CSS, and JavaScript

- Introduction to HTML5, CSS3, and JavaScript
- Understanding the Document Object Model (DOM)

Task**:** Build a simple static HTML page with basic CSS styling and JavaScript event handling.

References:

- HTML CSS JS : https://www.w3schools.com/
- Laravel official website: https://laravel.com/
- Vue.js official website: https://vuejs.org/
- Laravel documentation: https://laravel.com/docs
- Vue.js documentation: https://vuejs.org/v2/guide/
- Laracasts (Laravel video tutorials): https://laracasts.com/
- Vue Mastery (Vue.js video tutorials): https://www.vuemastery.com/
-

Day 2: Creating HTML5 and CSS3 layouts

- Basic HTML tags and attributes
- Basic CSS properties and selectors
- Creating simple layouts with HTML and CSS

Task**:** Build a simple responsive webpage layout using HTML and CSS.

Day 3: Basic JavaScript concepts and syntax

- Variables, data types, and operators
- Control structures (if statements, loops)
- Functions and event handling

Task**:** Build a simple JavaScript program that handles user input and displays output to the webpage.

Day 4: Introduction to jQuery

- What is jQuery?
- jQuery selectors and events
- Working with jQuery plugins

Task**:** Build a simple jQuery plugin that enhances the functionality of a basic webpage element.

Day 5: Introduction to PHP

- Introduction to PHP
- Setting up the development environment
- PHP syntax and data types

Task**:** Build a simple PHP script that handles user input and displays output to the webpage.

Day 6: Revision

2nd Week

Day 7: Installing Laravel

- Installing Laravel
- Understanding the Laravel directory structure
- Creating a simple Laravel application

Task**:** Set up a local Laravel development environment and create a simple Laravel application that displays data to the webpage.

Day 8: Laravel Routing

- Laravel Routing
- Creating routes in Laravel
- Named routes and route parameters

Task**:** Create custom routes in a Laravel application that correspond to specific pages or actions.

Day 9: Controllers in Laravel

- Controllers in Laravel
- Creating and using controllers in Laravel
- Passing data to view

Task**:** Create a custom controller in a Laravel application and pass data from the controller to the view.

Day 10: Introduction to RESTful APIs

- What is a RESTful API?
- Principles of RESTful design
- HTTP methods and status codes

Task**:** Design a simple RESTful API that allows users to access and manipulate data from a Laravel application.

Day 11: Creating RESTful routes in Laravel

- Creating RESTful routes in Laravel
- Route model binding
- Creating resource controllers

Task**:** Create a set of RESTful routes in a Laravel application that allow users to access and manipulate data using HTTP methods.

Day 12: Revision

3rd Week

Day 13: Introduction to Views in Laravel

- Introduction to Views in Laravel
- Creating and rendering views
- Blade templating syntax

Task**:** Build a simple Blade template in a Laravel application that displays data dynamically using templating syntax.

Day 14: Blade Templating

- Blade templating
- Using conditionals and loops in Blade

- Blade layouts and partials

Task**:** Build a custom Blade layout in a Laravel application that can be reused across multiple pages.

Day 15: Working with views and Blade templates in Laravel

- Passing data to views in Laravel
- Blade components and slots
- Extending Inheritance
- Asset management in Laravel

Task**:** Use Blade components and slots to create reusable UI components in a Laravel application and manage assets (CSS, JavaScript) in the application.

Day 16: Introduction to Vue.js

- Introduction to Vue.js
- Vue.js instance and lifecycle
- Vue.js data and methods

Task**:** Build a simple Vue.js application that displays data dynamically using Vue.js

Day 17: Vue.js templates and directives

- Vue.js templates and directives
- Vue.js conditional rendering and loops
- Handling user input with Vue.js events

Task**:** Build a custom Vue.js template that uses directives to conditionally render elements and handles user input with event handling.

Day 18: Revision

4th Week

Day 19: Vue.js components

- Vue.js components
- Creating and using Vue.js components
- Passing data to child components

Task**:** Create a custom Vue.js component that displays data passed from a parent component.

Day 20: Vue.js computed properties and watchers

- Vue.js computed properties

- Vue.js watchers
- Using computed properties and watchers for reactive data

Task**:** Create a Vue.js component that uses computed properties and watchers to display reactive data.

Day 21: Vue.js routing

- Introduction to Vue.js routing
- Creating routes in Vue.js
- Route parameters and navigation guards

Task**:** Create custom routes in a Vue.js application that correspond to specific pages or actions and use navigation guards to protect routes.

Day 22: Vue.js state management with Vuex

- Introduction to Vuex
- Vuex state, mutations, actions, and getters
- Updating state in a Vuex store

Task**:** Implement a simple Vuex store in a Vue.js application to manage application state.

Day 23: Integrating Laravel and Vue.js

- Integrating Laravel and Vue.js
- Using Axios to make API requests in Vue.js
- Passing data between Laravel and Vue.js

Task**:** Integrate a Vue.js frontend with a Laravel backend to create a full-stack application that displays data from a Laravel API.

Day 24: Revision

5th Week

Day 25: Authentication and Authorization in Laravel

- Introduction to Laravel authentication
- User registration and login
- User roles and permissions

Task**:** Implement user authentication and authorization in a Laravel application using the built-in Laravel authentication system.

Day 26: Creating an API in Laravel

- Creating an API in Laravel
- Securing an API with middleware
- API documentation with Swagger/OpenAPI

Task**:** Create a RESTful API in a Laravel application that allows users to access and manipulate data securely.

Day 27: Deploying a Laravel-Vue.js application

- Deployment options for Laravel-Vue.js applications
- Deploying to a shared hosting environment
- Deploying to a cloud hosting environment

Task**:** Deploy a Laravel-Vue.js application to a shared hosting environment or a cloud hosting environment.

Day 28: Project work and review

- Project work time for interns to work on a simple application
- Code review and feedback from mentors and senior developers
- Addressing questions and concerns from interns

Task**:** Work on a simple application project and receive feedback and guidance from mentors and senior developers.

Day 29: Project work and review

- Project work time for interns to work on a simple application
- Code review and feedback from mentors and senior developers
- Addressing questions and concerns from interns

Task**:** Finalize and present the simple application project to mentors and senior developers for final feedback and evaluation.

Day 30: Revision

6th Week

Day 31: Final project presentations

- Final project presentations from each intern
- Feedback and constructive criticism from mentors and senior developers

Task**:** Present the final project to mentors and senior developers and receive feedback and evaluation.

Day 32: Wrapping up the internship

- Final Q&A session with mentors and senior developers
- Sharing resources for further learning and development
- Providing feedback and evaluation to the interns

Task**:** Attend a final Q&A session with mentors and senior developers and receive feedback and evaluation for the internship.

Day 33-36: Optional workshops and further learning

- Optional workshops for interns to explore new technologies or deepen their understanding of certain topics
- Further learning opportunities and resources provided to interns for continued development

  Task**:** Attend optional workshops or explore new technologies or topics of interest to deepen understanding and continue working on real time project.