A.Need to know

1. Ecommerce
2. Employee Training Module.
3. Internet Banking
4. Online Ticketing
5. International Payment Gate Integration
6. Bangladeshi Local Payment Gateway Integration

Php Fundamental

1. Php founder ,name ,definition
2. Function && method:: function are all but method is declared under class or object with function keyword.
3. Var\_dump(), -> check data type &&& data length
4. Php data-type definition.8 (String, Integer, Float, Boolean, Array, Object, Null, Resource.)
5. Primitive data type 6 : String, number, Boolean, Null, undefined, symble . this primitive is not an object has no method)
6. Veriable vs constance.
7. Php operators (7 types)
8. Comparison operator {(==, <, >) Boolean (value) check} it’s called equality operator, {(===) data type & Boolean check} it’s called identical operator. && union operator (+)
9. Logical operator {(and/&&) all true}, {(or/||)one is true} condition, {Inequality operator(!==) not true}
10. String operator ( ‘.’ , ‘.=’ ), array operator ( ‘+ ‘).
11. Do while loop [ ($x= 2) do{return value ; $x++;} while ($x < = 10); ]
12. If condition [ if (condition) {execution} else{}; ]
13. For loop [ for($i=0; condition; $i++){execution};]
14. For each loop [foreach ($arrays as $Each value){code…}]
15. For each loop [foreach ($arrays as $key = > $val){code…}]
16. Array type : Indexed array, associative array(Key => Val), multi-dimensional array;
17. Super Global variable: $GLOBALS, $\_SERVER, $\_REQUEST, $\_POST, $\_GET, $\_FILE, $\_ENV, $\_SESSION, $\_COOCKIES,
18. $\_SERVER [‘’ PHP\_SELF’’], =>file path , $\_SERVER [‘’ SERVER\_NAME’’] => url path , $\_SERVER [‘’ SERVER\_ADDR’’] => server ip address, $\_SERVER['HTTP\_USER\_AGENT']=> get the browser properties.
19. Form Control php :

<form action=”<?php echo $\_SERVER[‘PHP\_SELF’]?>” method =”post” htmlspecialchars >

<input type=”text” name=”name” required></form>

1. If (isset($\_POST [‘’ submit’’])){
2. $input\_name = $\_POST [‘’ name’’];
3. $input\_email = $\_POST [‘’ email];
4. $input\_age = $\_POST [‘’ age’’]; }
5. Input field: $\_SERVER [‘’ REQUEST\_METHOD ]== ‘’POST’’
6. $name = validate($\_POST [‘’ input name’’]);
7. Echo $name;
8. Filter validation : filter\_var($\_POST [‘’ url’’], FILTER\_VALIDATE\_URL);
9. move\_uploaded\_file($tmpname, ‘foldername/’. $origenalname,)
10. $origenalname, = $\_FILES[‘imagefield’],[imagename]
12. Php functions …

* count(), for length
* assort (), for ascending sort
* <? php include/require “file name.php”;?>
* <? Php session\_start ()

$\_SESSION [“user’] = username;

$\_SESSION [‘password’] = 123;

Session\_unset ();

Session\_destroy () ;>

* If(! isset($\_COOCKIES[cookie-name]))

{set cookie(name, value, path,expire,domain,secure,httponly) or die(msg)}

* filter\_var(‘’$ url’’, FILTER\_VALIDATE\_URL), filter\_var(‘’$ str’’, FILTER\_SANITIZE\_SRTING);ip,email
* php Error :
* error\_reporting :: Phpinfo(); display\_error(on); => php version show
* error\_reporting(E\_ERROR | E\_NOTIC); => error showing in window
* Explode(‘/’,$str); => to break string and form an array
* Implode (); => to break array to form string.
* Case: strtoupper,strtolower,ucfirst,ucowrds
* Trim (‘/’, variable); => to remove whitespace or irrelevant character.
* Array\_count\_value(); =>to count same values
* Array\_map (function(){}); => to combine a function and array to make & return new array.
* Array\_product (); => array multiplying.
* Arrays name(just name), Srtings Names, Super Global Veriable,
* Php filter Senitization,
* Mysql Fetch Functions
* Regular Expression signs
* File\_get\_content, File\_put\_content
* Tutorials(83 - 92): form handling with php && javascript .

…………………………………………………….\*\*\*\*\*\*\*\*\*\*\*\*………………………………………………………..

OOP (object-oriented programing)

1. Oop: reusable code, well organize code, class module & reusable objects-method of class
2. Constructor=> auto call/loaded after class object call
3. distractor()=> load destructor at the end of script
4. Inheritance: child class used parent class’s properties & method.

* subclass extends superclass(),

1. Access modifier : public, private, protected (can be used in child class)
2. Abstract class {} -> childClass extends abstract Class{},one abstract method should be declared but not implemented in abstractClass but only override many time in abstract derive Class. {do not make object for Abstract class}
3. Interface : Multiple class(interface A, interface C) can be inherit in single derive class,

* do not make object for interface A{},
* interface method have no body part, no properties can be declared
* class B implements A,C { derive class override interface methods}

1. Static method (when you in another class => className :: static method() ) & properties (when you in own class =>self:: static properties)

* there no need to make any object.
* Parent class A{ public static $property, public static method() {} };
* derive classB extends StaticClass A{ parentA :: static method()};

1. Late Static Binding: subclass use their superclass static method & properties, when static class inherit another class then self::method() replace by static::method() ,therefore static method/properties only used in parent class.
2. Using Traits in PHP : Make common function under traits, that used in multiple class

* Trait A{ public fuction test(){};}
* Class B{ use A;} ; classC{ use A;}

1. Type hinting: declare data type in method,to avoid error
2. Php namespaces: Namespace allows us to use the same function name or class name in different parts of the same program/php file without causing a name collision.

* Namespace A =>class test{}
* Namespace B =>class test{}
* Php file: $obj = new A\test(), $obj=new B\test();

1. Method Chaining: multiple method call at the same time for same class

* $obj= new A(); $obj->method1()->method2(); return object{this}.

1. Final key word : this keyword prevent extend/override class & method.
2. Polymorphism() : superclass extended to subclass & subclass has new properties also override superclass method.
3. Object can be pass method’s values : $value = new className(properties value)
4. Magic method => (\_\_get, \_\_set, \_\_call): Method no need to call by object, they autoload.

* Type Magic methods: \_\_set() \_\_get(), \_\_isset(), \_\_unset(), \_\_autoload()
* \_\_call(), \_\_toString(), \_\_invoke(),

1. superGlobal variable (\_\_GET, \_\_POST),
2. Invoke method: public function\_\_invoke(){} ;call invoke method: $obj() [object self call]
3. Conditional Functions: class\_exists, method\_exists, interface\_exists, trait\_exists
4. To check class: class\_exists(“className”)
5. To check method: methos\_exists(object,’methodName’ )
6. Autoloading: function \_\_autoload($Filename) {include ’dir/’. $Filename . ‘.php’ ;}
7. (Standard PHP Library – SPL) Spl\_autoload\_register(function($Filename){ include ’dir/ ‘ .filename. ’.php’})
8. Iteration same as foreachloop
9. Php function: instanceOf => identify subclass is or not to specific superclass.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. MySQLi connection: $db = new mysqli (‘host(localhost)’, ’user(root)’, ’dbname’, ’pass’);
2. Query : $sql = “SELECT \* FROM my\_database”;

$stmt = $db->query($sql);

1. $stmt = $db->prepare($sql);|| $stmt-> execute();|| $stmt->bind\_result();|| $stmt->fetch()
2. Mysqli fetch methods:
3. PDO Fetch methods: PDO::FETCH\_ASSOC

* PDO::FETCH\_ASSOC
* PDO::FETCH\_NUM
* PDO::FETCH\_OBJ

1. Query : $sql = “insert into database\_tbl(name, email) values(?, ? )”;

* $stmt = $db->prepare($sql) ||
* $stmt->bindParam( $name, $email );
* $stmt-> execute();

1. BLOB : binary large object(65kbyte data store , image)
2. PDO: php data object
3. $dsn = “mysql : host= localhost; dbname= database;”;

$user = “root”;

$pass = “”;

$pdo =new PDO ($dsn,$user,$pass);

$stmt = $pdo ->query($sql); $stmt->fatch();

1. Query:$sql =“insert into database\_tbl(name, email,skills) values(:name, :email, :skills )”;

$stmt = $pdo->prepare($sql) ;

$stmt->bindParam(“:name”, $name );

$stmt->bindParam(“:email”, $email );

$stmt->bindParam(“:skills”, $skills );

$stmt-> execute();

OR

$stmt = $pdo->prepare($sql) ;

$stmt-> execute(array (:name => $name, :email =>$email , :skills => $skills));

$stmt->fatchAll();

OR

$stmt->bindaValue (( :name, :email, :skills : ‘afif’, $email, ‘java’);

$stmt-> execute();

1. Query : $sql = “insert into database\_tbl(name, email,skills) values(?,? )”;

$stmt = $pdo->prepare($sql)

$arr = array($name, $email, $skills)

$stmt-> execute($arr);

Php Myadmin

* select statements
* list of constraints: NOT NULL, UNIQUE, DEFAULT, CHECK, PRIMARYKEY.FOREINGKEY
* foreign key create: If your foreign key was formulated with ON DELETE CASCADE, the data in dependent tables will be removed automatically.
* AND OR NOT operator: Where age >=18 AND age<=30 ; all condition should true;

Where age >=18 OR age<=30 ; one condition is require to true;

Where NOT age >=18 ;only data show when condition is true;

* IN operator: Where age IN(18, 21); shorten the query, effective the OR operator
* Rang operator BETWEEN AND,
* Search operator: WHERE name LIKE “%s%”; “s%”; “%s” (0,1, multiple charecter)

WHERE name LIKE “\_s\_”; (only 1 charecter)

WHERE name LIKE “a%m”; “\_a%”(second character is a)

* Regular expression: begginign(^); end($); any char[wer];match rang[a-z];any one(p|2|d)
* DISTINCT operator: SELECT DISTINCT name FROM table\_2;(common one select);
* LIMIT OFSET operator
* COMMIT ROLEBACK operator:
* inner joint: SELECR customer.fristName, order.orderNumbr

FROM customer

INNER JOINT order

ON customer.id = order.customerId

* Subquery: SELECR fristName FROM customer WHERE id IN (SELECT age FREOM bacup WHERE age => 30);Nested query; strong or weack??
* GROUP BY : Group by operator is used with SELECT statement and Aggregate function to group rows together by common column values.
* HAVING CLAUSE: having clause always used with GROUP BY, SELECT statement.
* IF Clause:
* CASE Clause:
* String Functions:
* ALTER Command: Add cullomn, change data type,Delete column, modify table column
* Truncate vs Drop vs Delete: Drop Table->delete table, Truncate->remove data from table but not whole table.
* Aggregate function: avg(), sum(), min(), max(), count(),CONCATE(), Now()->current Time
* SELECT LAST\_INSERT\_ID() will return the last value assigned by the auto\_increment function.

\*Services Provider( provide all service by autoload to the app) / Services(any service from service container used by dependency injection in any class or controller ) container(laravel core service managed/or resolved in this container) dependency injecton (handle by other class constructor/ setter merhod/ interface)/ facade(nonstatic method can be call by facade to form static method) /traits(helper function)

\*\* Laravel Ajax crud

\*\* Rest Api in laravel (LP & SM)

\*\*API with JWT/Passport Auth [selected]

\*\* Json data fetch by Laravel Http::get/post/put/delete(): ( sarfside- 6)

\*\* React/Vue crud & restAPI with laravel

★ The Basic :

\* Routing:

redirect, route group, route parameters validation with Regx, model binding, callback route( default last route), Rate limiting route( limit for route calling), CORS, route cashing

\* Middleware: global Middleware, Route Middleware, group Middleware, Middleware paremeters, Terminal Middleware

\* controller: namespace, single action, controller middleware, resourch controller ( partial, api, nested, nameming), dependency injection (service class inject in many controller handle by service container)

\* request(method:get, post, put,patch, head, option, delete), http clients, dd($request);

\*response (redirect:name route, domain, controller method, flash messenge; view, json,file)

\* validation: rule, custom msg, custom rule

\* error handling: thrown exeption & user friendly custom exception handler create

\* collection: methods

\* helpers: custom helper

\* sessions : put,get,exist,has,flash,forget

★ Security : Email verification

★ Database: query builder, migration ★Elequent :api resources, relations, mutator/accessor/casts, cache

★ Jobs & Queues: Mail, Notification Queue

\*\* my php laravel & Accessories note

\*\* goggle drive note

\*\* note all documents & review all

\*\*\*\*\*\*\*\*----\*\*\*\*\*\*---\*\*\*\*\*\*\*---\*\*\*\*\*\*--\*\*\*\*\*\*\*\*

::. Advance Topic when used then learn::

★Deeging Deeper:

\* Events

\*Artisan

\* Broadcast

\* Rate limiting

\* Task scheduler

★Testing: http Tests(php unit tests)

★ Packaging

\* cachier(Stripe), \*Socialite, \*Telescope \* Sail, \* Sanctum

★Github

Javascript (Yahoo Baba)

* Variables: Const; const value cannot be redeclared
* Let value can be change multiple times, but not be declared again
* Var value can be change multiple times, and also be declared again
* Assignment operator: +=,-=,\*=, /=, %=, \*\*=
* Comparison operator: ==, ===(Value & Data type), <, >, <=, >=, !=, !==,
* Logical operator: &&(all condition are true), ||(between one is true), (not)!
* Ternary operator: {(condition) ? ‘true statement’: ‘ false statement’’}
* Alert box, confirm box, prompt box->input field
* Global variable: variable declared outer from a function, and value can be print in and out from this function.
* Local variable: variable declared in a function and only print inside value,
* Javascript Events: click, mouse hover, mouse->up, out, down, load, unload, scroll
* LOOP: initialize, condition, increment/decrement
* For/in loop: this loop used with OBJECT
* Foreach: this loop used with ARRAY
* Do/While loop: var a = 10; do{statement print ; increment/decrement}while(a <=10)
* For loop: for(a=1 ; a>= 10; a++){statement print}
* Array Methods: sort(), reverse(), pop(), push(), concat(), join(), slice(), splice(), isArray(), indexOd(),lastIndexOf(), every(), filter(),find(), fill(), includes(), some(), forEach(), toSring(), valueOf().map(),
* forEach loop: a= [1,2,3]; a.forEach(functon () { statement});
* String methods: length, toLowerCase(), toUpperCase(), startWith(), match(), replace(), repeat(), trim(), concat(), split(), slice(), toSrting(), valueOf().
* Number Methods: Number(), parseInt(), parseFloat(), isInteger(), toPrecision(x)
* Math methods:
* Date methods: toDateString(), getDate(), getFullMonth(), getDay(), setDate(), setFullYear(), setHours(),
* JS get methods: innerText, innerHtml, getAtrributes(),
* JS set method: innerText, innerHtml, setAtrributes(), removeAtrribute()
* querySelector: document.querySelector(“#id”), document.querySelectorAll(‘.class’)
* css styling methods: styte, className, classList
* ClassList Methods: classList.add(class1, class2,..), remove(class1, class2,..), toggle(class), contains(class), item(index), length
* Target element: parentElement, firstChild, nextSibling
* replaceChild(), removeChild(),
* Append Method: appendChild()->after all element, insertBefore() ->before all element, inserAdjusentElememt(), inserAdjusentHtml(), inserAdjusentText(),
* Find element: contains()->find child,grandchild anything ->true/false
* Form event: input(), change(), submit(), select(), invalid()

jQuery (Yahoo Baba)

* $(document).ready (): Document->selector; ready()->method
* Advance selector: $(p:first), $(p:last), $(ul il), $(li:odd); $(li:even) etc…
* Form event:
* Window event:
* Get/Set methods: text(),html, attr(), val()
* Css class methods: addclass(), removeClass(),toggleClass(),

$(selector).css(‘property’ : ’value’);

* Css multiple value: $(selector).css({‘property’ : ’value’ , ‘property’ : ’value’});
* ON method: $(‘#abc’).on({click:function(){}, mousehover:function(){}});
* OFF event: $(‘#abc’).off(‘click’, ‘mousehover’)=>on event are off
* Append & Prepend: prepend()->before element, append()->after all element
* Empty()->empty the target element, Remove()->remove target element
* reaplaceWith()->replace tags, &textNode, replaceAll()->only replace tags
* Height & width methods: width(), height(), inner/outerWidth/Height()
* Position & offset method: position(),offset(),
* hasClass()->true/false: to find particular class in DOM
* hide(), show(), toggle(),
* FadeIn(), FadeOut(), FadeTo(), FadeToggle(),
* Animate method: $(selector).animate({params}, speed, callback).
* Stop() ->stop animation
* Method Chaining: $(selector).click().slideUp().fadeIn();
* Ancestor methods: parent(), parents(), parentUntil(), offsetParent(), closest()
* Descendant method: children(), find()
* Sibling methods: sibling(), prev() , prevAll(), prevUntil(), next(), nextAll(), nextUntill().
* Filtering methods: first(), last(), eq(), not(), slice(), filter();
* Has(), Is():
* Each Method:
* Event objects: event.pageX , event.which, event.type, event.target, event.preventDefault(), event.stopPropagation(),
* event.data:

Github ()

React js (codding addict)

* React definition:
* NPM Basics commands
* Nested Components
* Props
* Props Destructuring // Props – Children || Spread Operator
* UseState – string, Array, object, number
* Show/Hide Component || ternary operatopr
* useEffect – Basics : useEffect function run a call back function and by default it run after every render. useEffect( () => {},[] ); array is the second perameter.
* useEffect use outside the components when using –eventLestening,dataFatching, signing up
* filte(), find(),map(),
* Event can be: onClick(function name |function with argument |arrow-function that return value/function called)
* Return always values of Object cant be key of object
* Array return : array.map((single name)=>return jsx with {Object-value} )
* form, subscription.
* useReducer
* useRef, useMemo, useCallback,

{Apply date on web development : 20/9/21}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dear Sir,

I am the Candidate who is passionate about technology, constantly seeking to learn advance tech and looking for a challenging role working on exciting projects in a great working environment. I am confident that I will be the right fit for the job.

My Skills on web development:

* **PHP OOP**
* **Php Framework (Laravel)**
* **Vue.js frontend**
* **React.js**
* **WordPress (CMS), ELEMENTOR page builder.**
* JavaScript, JavaScript ES6, jQuery, Ajax
* **Build RESTful APIs with JWT and Passport Authentication**
* Familiar with version control system - Git,
* MySQL (Database),
* HTML & CSS, Bootstrap4,
* Exception handling
* Halper function create (helper, trait, dependency injection)
* Jobs queues & Mailing
* Website testing (php unit testing + manual aggressive testing)
* Package Integration in Laravel
* Online Payment method (PayPal, Stripe, 2Checkout)
* Socialite in Laravel (Facebook, Twitter, Google)

Thank you for your time and consideration. I look forward to speak about this opportunity.

Best regard,

Nusrat Bhuiyan

Email: nusratdevo@gmail.com

Phone: 01686608828

Portfolio: <http://hr-nusrat.com>

Own Work: http://smartifybd.hr-nusrat.com/

Own Work: http://phpshop.hr-nusrat.com

Own Work: http://jsecom.hr-nusrat.com

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dear Sir,

It is very exciting to see the opening at Brain Station 23 Ltd. I heard more about the reputation of the company and this would be the golden opportunity to me to be a part of your intellectual dev team

Dear Sir,

I am writing this letter to express my interest in interviewing for the position of Web Developer in **PHP(Laravel) + vue.js.** Since I have 1.5+ years of experience on fiver and local-market in frontend and backend Web developing services. In addition, I actively seek out new technologies and stay up-to-date on new trends and advancements. I have a copy of my resume detailing my experience along with links to own work.

In addition to my knowledge base, I actively seek out new technologies and stay up-to-date on new trends and advancements. Therefore this internship opportunity will be the best path to joint techy group like to for my dream comes true. I send my resume detailing my experience along with links to websites. Thank you.

Best regard,

Nusrat Bhuiyan

Email: hr.nusratdevo@gmail.com

Phone: 01686608828

Portfolio Tips

1. Have a presentable website 2. Good hosting and domain 3. Keep it simple 4. Don't exaggerate your skill set. Showing 2/3 deep advance knowledge but overall basic in other skills. 5. Have live examples/demos 3/6. Be selective of projects (not to include small project but presentable looking 2/3 deep skill project) 7. Customize course projects 8. Easy contact 9. Try to have web presence and brand 10. Be yourself. Be real.