

Laravel Notes

STEP 0: Install Laravel Project

```
// Create new Laravel project
composer create-project laravel/laravel [project-name]

// Alternative Create project using installer
laravel new [project-name]

// Navigate into project folder
cd [project-name]
```

STEP 1: Start XAMPP & Create Database

Start Services:

- Open XAMPP Control Panel
- Start Apache
- Start MySQL

```
// Access MySQL
mysql -u root -p
// Press Enter if no password
```

```
// Or directly:
mysql -u root
```

```
// Create database
CREATE DATABASE [db_name];
```

```
// Exit MySQL
EXIT;
```

STEP 2: Configure Environment File

Find and modify these lines:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=[db_name]
DB_USERNAME=root
DB_PASSWORD=
```

STEP 3: Create Migration with Table

```
php artisan make:migration [migration_name] --create=[tbl_name]
```

What happens:

- Creates migration file in `database/migrations/`
- File name: `YYYY_MM_DD_HHMMSS_migration_name.php`
- Pre-filled with table structure

STEP 4: Define Table Columns in Migration

Example for tbl_user (Billing)

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('tbl_user', function (Blueprint $table) {
            // Primary Key
            $table->id('ID');

            // UserInformations
```

```

        $table->string('firstName');
        $table->string('lastName');
        $table->string('midInitial');
        $table->string('Email');
        $table->bigInteger('Contact');
        $table->string('Address');

        // Timestamps
        $table->timestamps();
    });
}

/**
 * Reverse the migrations.
 */
public function down(): void
{
    Schema::dropIfExists('tbl_user');
}
};

```

STEP 5: Run Migration to Create Table

```
php artisan migrate
```

STEP 6: Create Model

```
php artisan make:model [model_name]
```

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Mobilebill extends Model
{

```

```
use HasFactory;

protected $table = 'tbl_user';
protected $primaryKey = 'ID';

protected $fillable = [
    'firstName',
    'lastName',
    'midInitial',
    'Email',
    'Contact',
    'Address',
];
}
```

STEP 7: Create Controller

```
php artisan make:controller UserController
```

```
<?php

namespace App\Http\Controllers;

use App\Models\Mobilebill;
use Illuminate\Http\Request;

class UserController extends Controller
{
    public function index()
    {
        return view('billing');
    }

    public function calculate(Request $request)
    {
        $validated = $request->validate([
            'firstName' => 'required|string|max:100',
            'lastName' => 'required|string|max:100',
            'midInitial' => 'required|string|max:10',
        ]);
    }
}
```

```

        'Email' => 'required|email|max:100',
        'Contact' => 'required|numeric|digits_between:10,15',
        'Address' => 'required|string|max:200',
    ]);

    // Save to DB
    Mobilebill::create([
        'firstName' => $request->firstName,
        'lastName' => $request->lastName,
        'midInitial' => $request->midInitial,
        'Email' => $request->Email,
        'Contact' => $request->Contact,
        'Address' => $request->Address,
    ]);

    return view('view', [
        'firstName' => $request->firstName,
        'lastName' => $request->lastName,
        'midInitial' => $request->midInitial,
        'Email' => $request->Email,
        'Contact' => $request->Contact,
        'Address' => $request->Address,
    ]);
}
}

```

STEP 8: Create Routes

```

use App\Http\Controllers\UserController;

Route::get('/telecom/billing', [UserController::class, 'index']);
Route::post('/telecom/calculate', [UserController::class, 'calculate']);

```

STEP 9: Create View

Form View:

```

// view.blade.php
<!DOCTYPE html>

```

```

<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Telecom Billing Form</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body class="bg-light">
<div class="container mt-5">
  <div class="card shadow p-4">
    <h2 class="mb-4 text-center">Telecom Billing Form</h2>

    @if ($errors->any())
      <div class="alert alert-danger">
        <ul class="mb-0">
          @foreach ($errors->all() as $error)
            <li>{{ $error }}</li>
          @endforeach
        </ul>
      </div>
    @endif

    <form action="/telecom/calculate" method="POST">
      @csrf

      // Input Form Example:
      <div class="row mb-3">
        <div class="col-md-4">
          <label class="form-label">First Name</label>
          <input type="text" name="firstName" class="form-control" required>
        </div>
        <div class="col-md-4">
          <label class="form-label">Middle Initial</label>
          <input type="text" name="midInitial" class="form-control" maxlength
="5">
        </div>
        <div class="col-md-4">
          <label class="form-label">Last Name</label>
          <input type="text" name="lastName" class="form-control" required>
        </div>

```

```

</div>
<div class="row mb-3">
  <div class="col-md-6">
    <label class="form-label">Email</label>
    <input type="email" name="Email" class="form-control" required>
  </div>
  <div class="col-md-6">
    <label class="form-label">Contact Number</label>
    <input type="number" name="Contact" class="form-control" required>
  </div>
</div>
<div class="mb-3">
  <label class="form-label">Address</label>
  <input type="text" name="Address" class="form-control" required>
</div>

// Radio Button Form Example:
<div class="row mb-3">
  <div class="col-md-6">
    <label class="form-label">Data (GB)</label>
    <input type="number" name="Data_GB" class="form-control" required>
  </div>
  <div class="col-md-6">
    <label class="form-label d-block">Plan Type</label>
    <div class="form-check form-check-inline">
      <input class="form-check-input" type="radio" name="Plan_Type" id="prepaid" value="Prepaid" required>
      <label class="form-check-label" for="prepaid">Prepaid</label>
    </div>
    <div class="form-check form-check-inline">
      <input class="form-check-input" type="radio" name="Plan_Type" id="postpaid" value="Postpaid">
      <label class="form-check-label" for="postpaid">Postpaid</label>
    </div>
    <div class="form-check form-check-inline">
      <input class="form-check-input" type="radio" name="Plan_Type" id="business" value="Business">
      <label class="form-check-label" for="business">Business</label>
    </div>
  </div>
</div>

```

```

    </div>

    // Checkbox Form Example:
    <div class="form-check mb-2">
        <input type="checkbox" name="roaming_checkbox" class="form-check-input" id="roaming">
        <label for="roaming" class="form-check-label">Include Roaming Fee (₦750)
    </label>
    </div>

    <div class="form-check mb-4">
        <input type="checkbox" name="overdue_checkbox" class="form-check-input" id="overdue">
        <label for="overdue" class="form-check-label">Apply Overdue Charge (2
    5%)</label>
    </div>

    <button type="submit" class="btn btn-primary w-100">Calculate Bill</button>
    </form>
</div>
</div>
</body>
</html>

```

Display View:

```

// display.blade.php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Billing Summary</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
    rel="stylesheet">
</head>
<body class="bg-light">
<div class="container mt-5">
    <div class="card shadow p-4">
        <h2 class="text-center mb-4">Billing Summary</h2>

```



```

<table class="table table-bordered">
  <tbody>
    <tr><th>Full Name</th><td>{{ $firstName }} {{ $midInitial }}. {{ $lastName }}
</td></tr>
    <tr><th>Email</th><td>{{ $Email }}</td></tr>
    <tr><th>Contact</th><td>{{ $Contact }}</td></tr>
    <tr><th>Address</th><td>{{ $Address }}</td></tr>
    <tr><th>Plan Type</th><td>{{ $Plan_Type }}</td></tr>
    <tr><th>Data (GB)</th><td>{{ $Data_GB }}</td></tr>
    <tr><th>Data Charge</th><td>{{ number_format($data_charge, 2) }}</td>
</tr>
    <tr><th>Roaming Fee</th><td>{{ number_format($roaming_fee, 2) }}</td>
</tr>
    <tr><th>Overdue Charge</th><td>{{ number_format($overdue_charge, 2)
}}</td></tr>
    <tr class="table-primary"><th>Total Bill</th><td><strong>{{ number_forma
t($total_bill, 2) }}</strong></td></tr>
  </tbody>
</table>
*
  <a href="/telecom/billing" class="btn btn-secondary mt-3 w-100">Go Back</a>
</div>
</div>
</body>
</html>

```

ADDITIONAL NOTES

```

// Add these to migrations
$table->integer('Data_GB');
$table->string('Plan_Type');

$table->double('Data_Charge');
$table->double('Roaming_Fee');
$table->double('Overdue_Charge');
$table->double('Total_Bill');

```

```

// Add to $fillable
'Data_GB',

```

```
'Plan_Type',  
'Data_Charge',  
'Roaming_Fee',  
'Overdue_Charge',  
'Total_Bill'
```

```
// Add to create db and redirect view
```

```
'Data_GB' ⇒ $request→Data_GB,  
'Plan_Type' ⇒ $request→Plan_Type,  
'Data_Charge' ⇒ $data_charge, //use the variable computed on the server side  
'Roaming_Fee' ⇒ $roaming_fee,  
'Overdue_Charge' ⇒ $overdue_charge,  
'Total_Bill' ⇒ $total_bill,
```

```
// You can add logic block within the controller
```

```
$rate = match ($request→Plan_Type) {  
'Prepaid' ⇒ 15.50,  
'Postpaid' ⇒ 12.75,  
'Business' ⇒ 10.25,  
default ⇒ 0  
};
```

```
data_charge = $request→Data_GB * $rate;  
$roaming_fee = $request→has('roaming_checkbox') ? 750 : 0;  
$overdue_charge = $request→has('overdue_checkbox') ? $data_charge * 0.25 : 0;  
$total_bill = $data_charge + $roaming_fee + $overdue_charge;
```