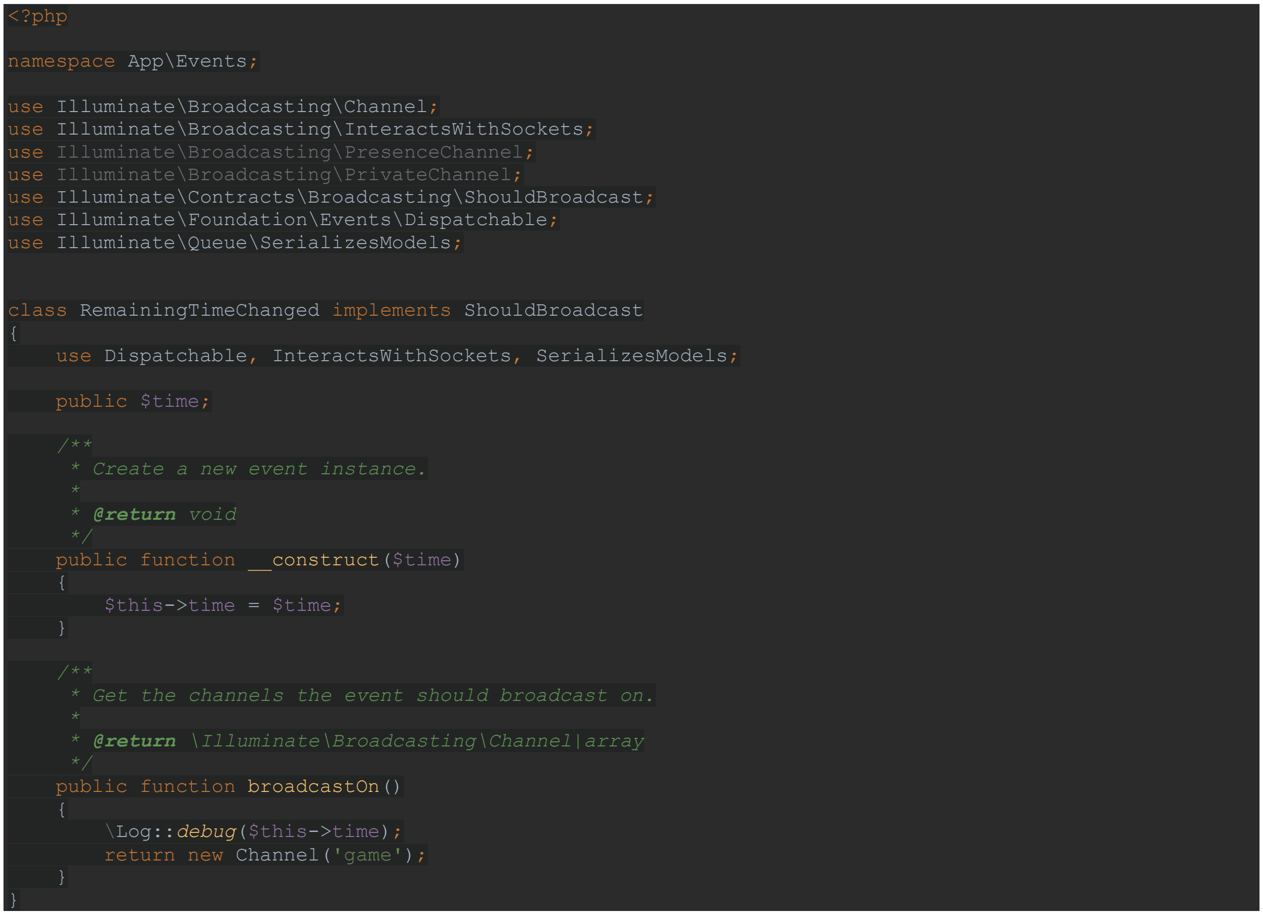
Step 0 : setup docx ta follow kore nite hobe.
Step 1 : Layout er design and script thik kore nite hobe.

```
@extends('layouts.app')

@push('styles')
    <style type="text/css">
        @keyframes rotate {
            from {
                transform: rotate(0deg);
            }
            to {
                transform: rotate(360deg);
            }
        }

        .refresh {
            animation: rotate 1.5s linear infinite;
        }
    </style>
@endpush

@section('content')
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-md-8">
                <div class="card">
                    <div class="card-header">{{ __('Game') }}</div>

                   <div class="card-body">
                       <div class="text-center">
                           <img src="{{ asset('img/circle.png') }}" alt="" id="circle" class=""
style="height: 250px; width: 250px">
                           <p id="winner" class="display-1 d-none text-primary"></p>
                       </div>

                       <hr>

                       <div class="text-center">
                           <label for="" class="font-weight-bold h5">Your Bet</label>
                           <select name="" id="" class="custom-select col-auto">
                               <option selected>Not In</option>

                               @foreach(range(1,12) as $number)
                                   <option>{{ $number }}</option>
                               @endforeach
                           </select>
                           <hr>
                           <p class="font-weight-bold h5">Remaining Time</p>
                           <p class="font-weight-bold h5 text-danger" id="timer">Waiting to start</p>
                           <hr>
                           <p id="result" class="h1"></p>
                       </div>
                   </div>
                </div>
            </div>
        </div>
    </div>
@endsection

@push('scripts')
    <script>


        // const circleElement = document.getElementById('circle');
        // const timerElement = document.querySelector('#timer');
        // const winnerElement = document.getElementById('winner');
        // const betElement = document.getElementById('bet');
        // const resultElement = document.getElementById('result');
        //
        //
        // Echo.channel('game').listen('RemainingTimeChanged', function(e) {
        //     timerElement.innerText = e.time;
        //     circleElement.classList.add('refresh');
        //     winnerElement.classList.add('d-none');
        //     resultElement.innerText = '';
        //     resultElement.classList.remove('text-success');
        //     resultElement.classList.remove('text-danger');
        //
        // }).listen('WinnerNumberGenerated', function(e) {
        //     circleElement.classList.remove('refresh');
        //
        //     let winner = e.number;
        //     winnerElement.innerText = winner;
        //     winnerElement.classList.remove('d-none');
        //
        //     let bet = betElement[betElement.selectedIndex].innerText;
        //
        //     if (bet == winner) {
        //         resultElement.innerText = "You Win";
        //         resultElement.classList.add('text-success');
        //     } else {
        //         resultElement.innerText = "You Lose";
        //         resultElement.classList.add('text-danger');
        //     }
        //
        // })

    </script>
@endpush
```

Step 2 : Duita event create kora holo, **RemainingTimeChanged** and **WinnerNumberGenerated**

RemainingTimeChanged code,



```php
<?php

namespace App\Events;

use Illuminate\Broadcasting\Channel;
use Illuminate\Broadcasting\InteractsWithSockets;
use Illuminate\Broadcasting\PresenceChannel;
use Illuminate\Broadcasting\PrivateChannel;
use Illuminate\Contracts\Broadcasting\ShouldBroadcast;
use Illuminate\Foundation\Events\Dispatchable;
use Illuminate\Queue\SerializesModels;


class RemainingTimeChanged implements ShouldBroadcast
{
    use Dispatchable, InteractsWithSockets, SerializesModels;

    public $time;

    /**
     * Create a new event instance.
     *
     * @return void
     */
    public function __construct($time)
    {
        $this->time = $time;
    }

    /**
     * Get the channels the event should broadcast on.
     *
     * @return \Illuminate\Broadcasting\Channel|array
     */
    public function broadcastOn()
    {
        \Log::debug($this->time);
        return new Channel('game');
    }
}
```

WinnerNumberGenerated code,

```php
<?php

namespace App\Events;

use Illuminate\Broadcasting\Channel;
use Illuminate\Broadcasting\InteractsWithSockets;
use Illuminate\Broadcasting\PresenceChannel;
use Illuminate\Broadcasting\PrivateChannel;
use Illuminate\Contracts\Broadcasting\ShouldBroadcast;
use Illuminate\Foundation\Events\Dispatchable;
use Illuminate\Queue\SerializesModels;


class WinnerNumberGenerated implements ShouldBroadcast
{
    use Dispatchable, InteractsWithSockets, SerializesModels;

    public $number;

    /**
     * Create a new event instance.
     *
     * @return void
     */
    public function __construct($number)
    {
        $this->number = $number;
    }

    /**
     * Get the channels the event should broadcast on.
     *
     * @return \Illuminate\Broadcasting\Channel|array
     */
    public function broadcastOn()
    {
        \Log::debug($this->number);
        return new Channel('game');
    }
}
```

Step 3: making a command - php artisan make:command GameExecutor

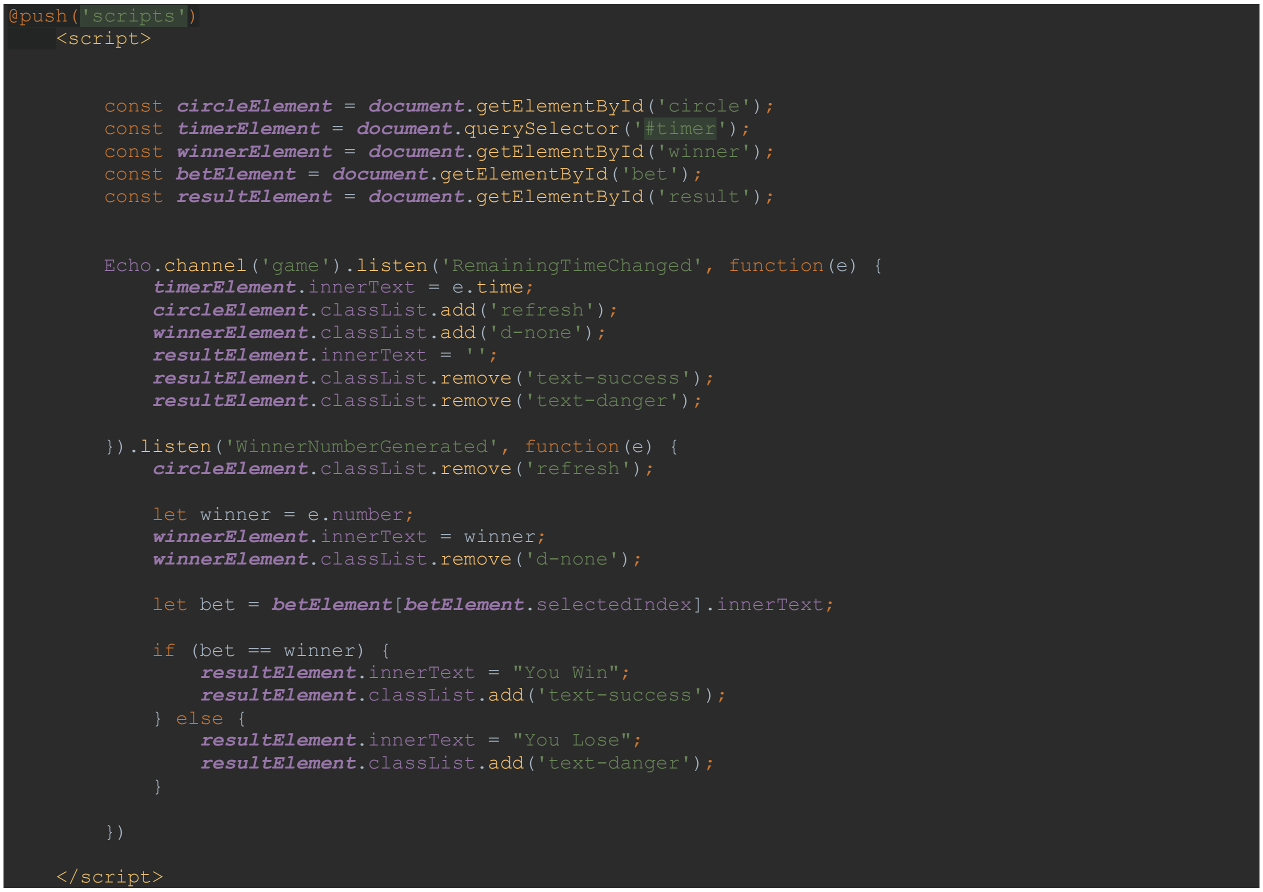
Full **GameExecutor code,**

```
<?php

namespace App\Console\Commands;

use App\Events\RemainingTimeChanged;
use App\Events\WinnerNumberGenerated;
use Illuminate\Console\Command;

class GameExecutor extends Command
{
    /**
     * The name and signature of the console command.
     *
     * @var string
     */
    protected $signature = 'game:execute';

    /**
     * The console command description.
     *
     * @var string
     */


    protected $description = 'Start executing the game';
    private $time = 15;

    /**
     * Create a new command instance.
     *
     * @return void
     */
    public function __construct()
    {
        parent::__construct();
    }

    /**
     * Execute the console command.
     *
     * @return int
     */
    public function handle()
    {
        while (true) {
            broadcast(new RemainingTimeChanged($this->time, ' s'));
            $this->time--;
            sleep(1);

            if ($this->time === 0) {
                $this->time = "Waiting to start";
                broadcast(new RemainingTimeChanged($this->time));
                broadcast(new WinnerNumberGenerated(mt_rand(1, 12)));
                sleep(5);
                $this->time = 15;
            }
        }

//        return 0;
    }
}
```

GameExecutor er moddhe je kaj ta kora hoise,

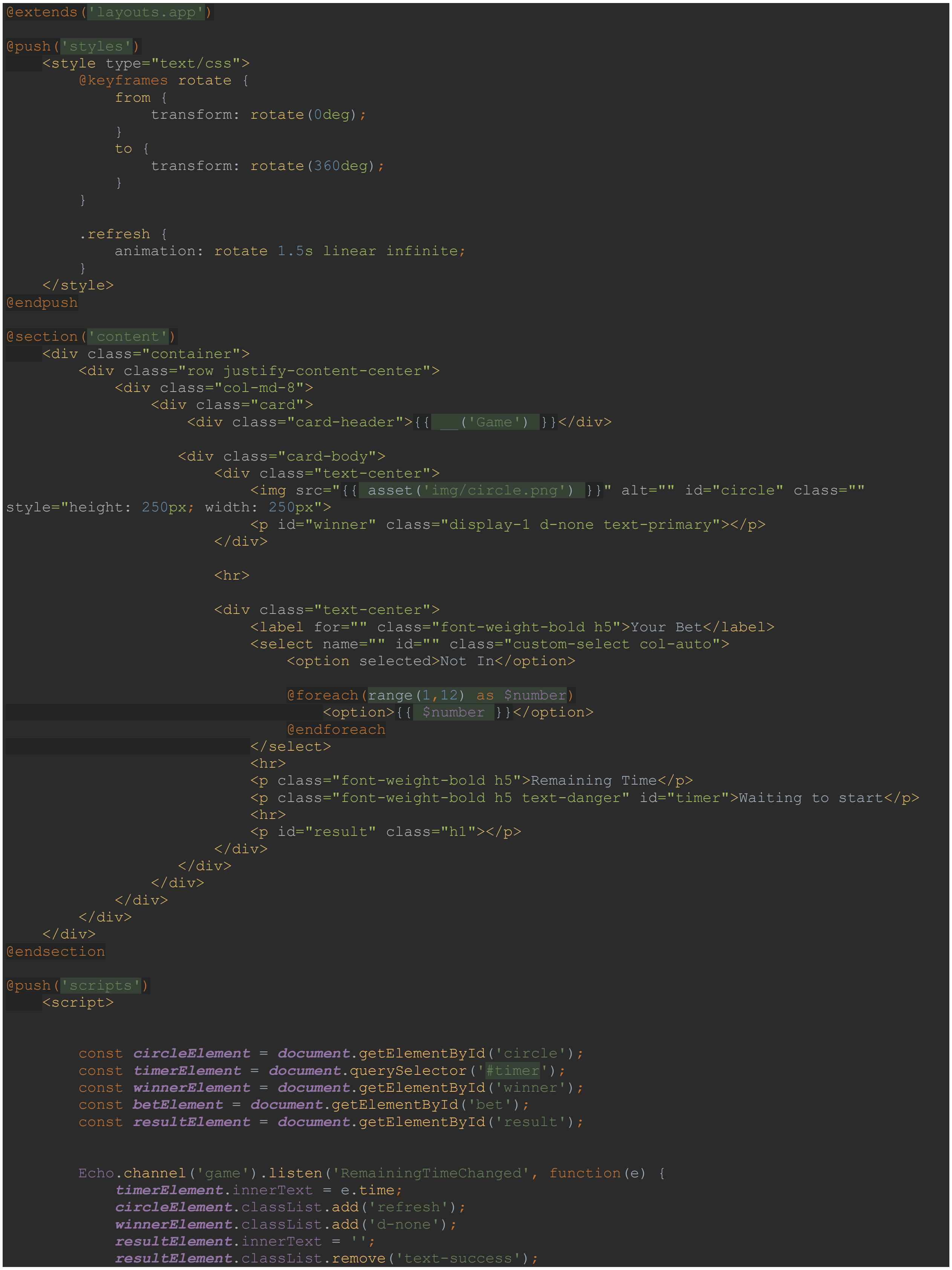
1. $signature er moddhe game:execute rakha hoise jate command likhlei run kore.
2. Time limit 15 sec dewa hoise.
3. Handle funcion er moddhe logic implement kora hoise
4. Sleep means seconds break

Then lastly script code add korte hobe,



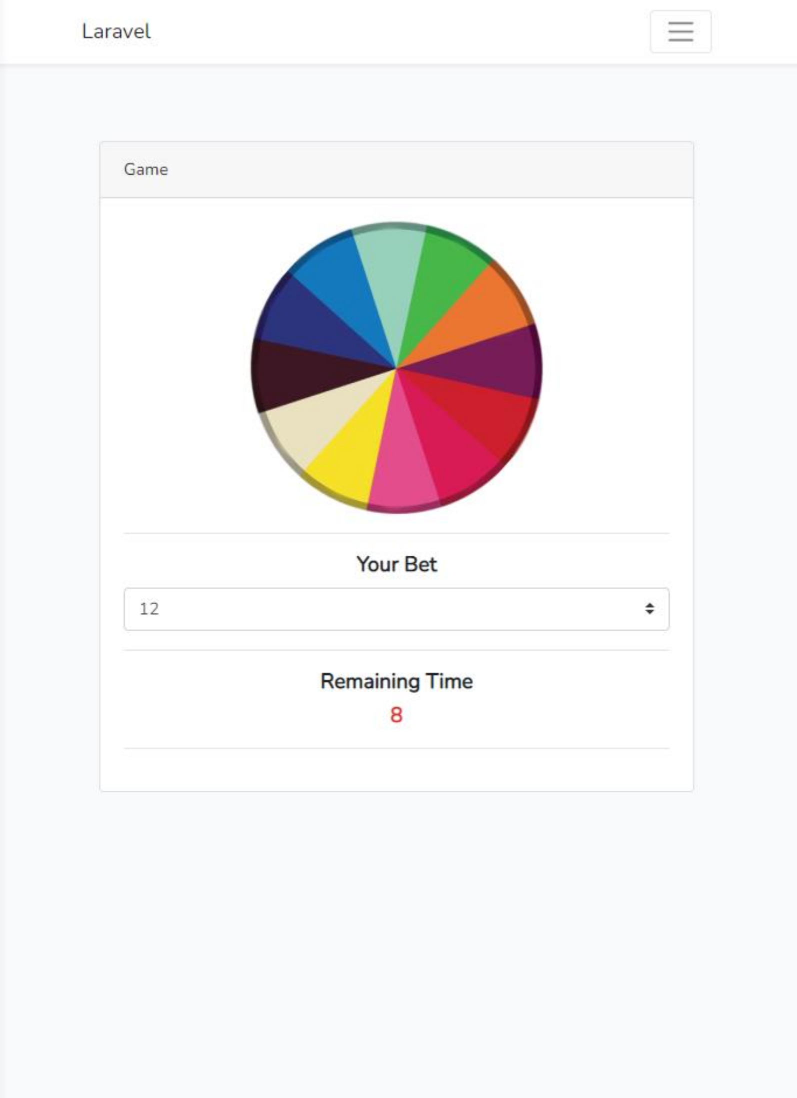
```
@push('scripts')
    <script>


        const circleElement = document.getElementById('circle');
        const timerElement = document.querySelector('#timer');
        const winnerElement = document.getElementById('winner');
        const betElement = document.getElementById('bet');
        const resultElement = document.getElementById('result');


        Echo.channel('game').listen('RemainingTimeChanged', function(e) {
            timerElement.innerText = e.time;
            circleElement.classList.add('refresh');
            winnerElement.classList.add('d-none');
            resultElement.innerText = '';
            resultElement.classList.remove('text-success');
            resultElement.classList.remove('text-danger');

        }).listen('WinnerNumberGenerated', function(e) {
            circleElement.classList.remove('refresh');

            let winner = e.number;
            winnerElement.innerText = winner;
            winnerElement.classList.remove('d-none');

            let bet = betElement[betElement.selectedIndex].innerText;

            if (bet == winner) {
                resultElement.innerText = "You Win";
                resultElement.classList.add('text-success');
            } else {
                resultElement.innerText = "You Lose";
                resultElement.classList.add('text-danger');
            }

        })

    </script>
```

Full **game->show.blade.php** code,



```
@extends('layouts.app')

@push('styles')
    <style type="text/css">
        @keyframes rotate {
            from {
                transform: rotate(0deg);
            }
            to {
                transform: rotate(360deg);
            }
        }

        .refresh {
            animation: rotate 1.5s linear infinite;
        }
    </style>
@endpush

@section('content')
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-md-8">
                <div class="card">
                    <div class="card-header">{{ __('Game') }}</div>

                   <div class="card-body">
                       <div class="text-center">
                           <img src="{{ asset('img/circle.png') }}" alt="" id="circle" class=""
style="height: 250px; width: 250px">
                           <p id="winner" class="display-1 d-none text-primary"></p>
                       </div>

                       <hr>

                       <div class="text-center">
                           <label for="" class="font-weight-bold h5">Your Bet</label>
                           <select name="" id="" class="custom-select col-auto">
                               <option selected>Not In</option>

                               @foreach(range(1,12) as $number)
                                   <option>{{ $number }}</option>
                               @endforeach
                           </select>
                           <hr>
                           <p class="font-weight-bold h5">Remaining Time</p>
                           <p class="font-weight-bold h5 text-danger" id="timer">Waiting to start</p>
                           <hr>
                           <p id="result" class="h1"></p>
                       </div>
                   </div>
                </div>
            </div>
        </div>
    </div>
@endsection

@push('scripts')
    <script>


        const circleElement = document.getElementById('circle');
        const timerElement = document.querySelector('#timer');
        const winnerElement = document.getElementById('winner');
        const betElement = document.getElementById('bet');
        const resultElement = document.getElementById('result');


        Echo.channel('game').listen('RemainingTimeChanged', function(e) {
            timerElement.innerText = e.time;
            circleElement.classList.add('refresh');
            winnerElement.classList.add('d-none');
            resultElement.innerText = '';
            resultElement.classList.remove('text-success');
            resultElement.classList.remove('text-danger');

        }).listen('WinnerNumberGenerated', function(e) {
            circleElement.classList.remove('refresh');

            let winner = e.number;
            winnerElement.innerText = winner;
            winnerElement.classList.remove('d-none');

            let bet = betElement[betElement.selectedIndex].innerText;

            if (bet == winner) {
                resultElement.innerText = "You Win";
                resultElement.classList.add('text-success');
            } else {
                resultElement.innerText = "You Lose";
                resultElement.classList.add('text-danger');
            }

        })

    </script>
@endpush
```

Output,



Your Bet

12

Remaining Time

8