



TO DO PALNNER

Task - 4



Chat Application and Website

- To create a to do planner

LMS Username	Name	Batch
2113a54271	SELCIYA DASS S	A54
2113a54265	SANDHIYA S	A54
2113a54293	KAVIYA G	A54
2113a54290	YASMIN I	A54



Task 4 :: Backend (Module 4)

Do database modelling and create models

- Design schema for all the data to be stored
- Start mongodb local server and point the backend to the server
- Create mongoose schema at the backend
- Run test queries to set up the database

Create Various APIs to ensure data flow within the website

- Define all the routes for the backend
- Add authentication middleware
- Add controllers for all the paths to handle api request
- Create Environment variables for all authentication keys
- Note: Always handle all possible cases with the request

Evaluation Metric:

100% Completion of the above tasks

Learning outcome

- Understanding Nosql databases modeling
- Querying and filtering mongodb
- Understanding various req methods
- Getting familiar with cookies
- Server side authentication

```
$(document).ready(function(){  
  
    var hour;  
    var timeInterval;  
    // header section  
    // display time and update every second  
    timeInterval = setInterval(function(){  
        // display day of week  
        $('#day-display').text(moment().format('dddd'));  
        // display time of day  
        $('#time-digits').text(moment().format('h:mm'));  
        // display am / pm for style purposes  
        $('#time-period').text(moment().format('a'));  
  
        // save time to variable  
        hour = parseInt(moment().format('H'));  
        var minutes = parseInt(moment().format('mm'));  
  
        // timeblocks  
        if (hour < 9){  
            $('time-block').removeClass("present past").addClass("future");  
        }  
    }, 1000);  
});
```

```
}  
  if (hour > 16){  
    $('time-block').removeClass("present future").addClass("past");  
  }  
  $( "div.time-block" ).each(function() {  
    var timeblockNum = parseInt($( this ).data("value"));  
    if (hour < timeblockNum){  
      $( this ).removeClass("past present").addClass("future");  
    }  
    if (hour == timeblockNum){  
      $( this ).removeClass("past future").addClass("present");  
    }  
    if (hour > timeblockNum){  
      $( this ).removeClass("present future").addClass("past");  
    }  
  });
```

```
// current time display
if (hour > 17 || hour < 9){
    $('#current-time').css("display", "none");
} else {
    $('#current-time').css("display", "grid");
}
// current time line position
var position = ((hour * 100) - 802) + (minutes * 1.65);
$('#current-time').css('top', position+'px');

}, 100);
```

```
// timeblocks object
var blocks = {
    block9: {
        time: "9AM - 10AM",
        title: "",
        description: ""
```

```
},  
  block10: {  
    time: "10AM - 11AM",  
    title: "",  
    description: ""  
  },  
  block11: {  
    time: "11AM - 12PM",  
    title: "",  
    description: ""  
  },  
  block12: {  
    time: "12PM - 1PM",  
    title: "",  
    description: ""  
  },  
  block13: {  
    time: "1PM - 2PM",  
    title: "",  
    description: ""  
  }
```

```
    },  
    block15: {  
      time: "3PM - 4PM",  
      title: "",  
      description: ""  
    },  
    block16: {  
      time: "4PM - 5PM",  
      title: "",  
      description: ""  
    }  
  }  
};
```

```
// LOCAL STORAGE
```

```
var storageKey = localStorage.getItem("storageKey");  
init();
```



```
function renderBlocks(){
    $( "div.time-block" ).each(function() {
        var timeblockNum = $( this ).attr("data-value");
        console.log(timeblockNum);
        console.log(this);
        // clear old text (need when saving)
        $( this ).empty();
        // add text to timeblocks
        $( this ).append($("<h4
class='title'>" + blocks["block"+timeblockNum].title + "</h4>"));
        $( this ).append($("<p
class='description'>" + blocks["block"+timeblockNum].description + "</p>"));
        if
(blocks["block"+timeblockNum].title!=="")||blocks["block"+timeblockNum].des
cription!==""){
            $( this ).removeClass("empty");
        }
    });
};
```

```
function init() {  
    // check if local storage has been used else get data from local storage  
    if(storageKey===null){  
        console.log("nothing in storage");  
    } else {  
        blocks = JSON.parse(localStorage.getItem("storageKey"));  
    }  
    // Render event text  
    renderBlocks();  
};  
  
function storeBlocks() {  
    // store timeblock objects in local storage  
    localStorage.setItem("storageKey", JSON.stringify(blocks));  
};
```

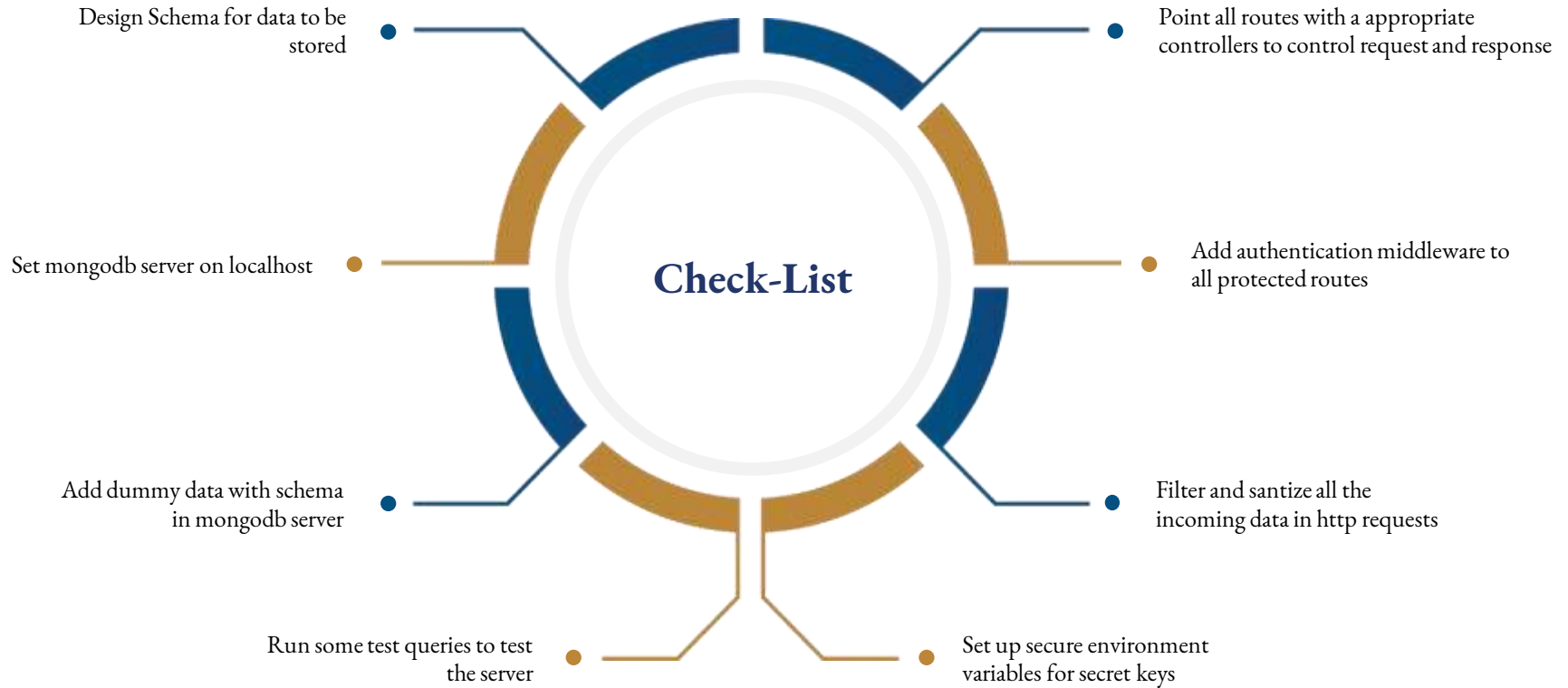
```
// declare variables for functions  
var blockNum;
```

```
// display form and info for selected time block
```

```
function addText(timeblockdiv){  
    // get the selected time block number from the div.time-block data-value  
    blockNum = ( $( timeblockdiv ).attr("data-value")).toString();  
    // display values for selected timeblock in form  
    $('#form-time').text(blocks["block"+blockNum].time);  
    $('#title').val(blocks["block"+blockNum].title);  
    $('#description').val(blocks["block"+blockNum].description) ;  
};
```

```
// EVENTS// timeblock on click display form
$('.time-block').on("click", function(){
  // display form
  $('#text-form').css('display', 'block');
  addText(this);
});
// close button
$('#close').on("click", function(){
  $('#text-form').css('display', 'none');
}); // save button
$('#save').on("click", function(event){
  event.preventDefault();
  // get values from form and add to blocks object
  blocks["block"+blockNum].title = $('#title').val();
  blocks["block"+blockNum].description = $('#description').val();
  storeBlocks();
  renderBlocks();
  // close form pop up
  $('#text-form').css('display', 'none');
});
}); // end document ready
```

Assessment Parameter



Submission Github



https://github.com/ReenuYasmin29/Naan_mudhalvan_fullstack

Thank you!

