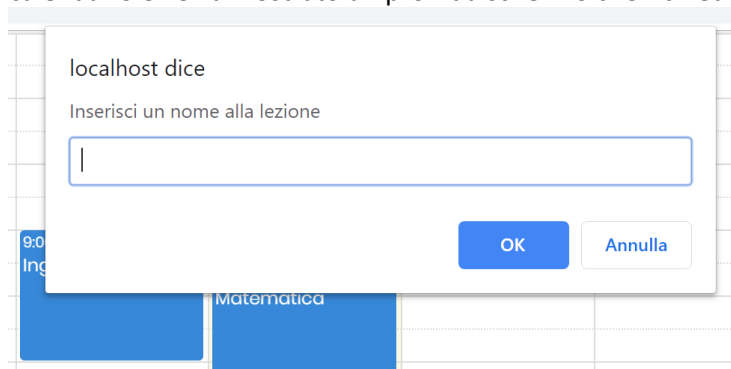


Diario di lavoro

Luogo	Scuola Arti e Mestieri Trevano
Data	15.11.2019

Lavori svolti

Durante la giornata di oggi ho implementato la funzione di aggiunta e l'ingrandimento delle lezioni direttamente dal calendario. Per aggiungere una lezione è sufficiente cliccare su una parte vuota del calendario e verrà mostrato un prompt a schermo che richiederà di inserire un titolo alla lezione:



Una volta inserito il nome della lezione essa verrà aggiunta al database e dunque salvata. Adesso da ora in poi si vedrà quella lezione sul calendario fino a quando non verrà eliminata dal database. Per fare ciò ho riempito il metodo "select" della libreria del calendario che mi verificherà la ora di inizio e di fine lezione:

Il metodo select:

```
select: function(selectionInfo) {
    var title = prompt("Inserisci un nome alla lezione");
    if(title) {
        if(checkName(title)){
            var start = moment(selectionInfo.start).format("Y-MM-DD HH:mm:ss");
            var end = moment(selectionInfo.end).format("Y-MM-DD HH:mm:ss");
            if(checkEvent(start, end) != null) {
                $.ajax({
                    url: "http://localhost/GestioneVacanze/Calendario/insert",
                    type: "POST",
                    data: {title: title, start: start, end: end, diff: checkEvent(start, end)},
                    success: function (response) {
                        if(response == "1"){
                            $.notify("Lezione Aggiunta", "success");
                        }else if(response == "2"){
                            $.notify("Giorno non disponibile per lezioni", 'error');
                        }else if(response == "3"){
                            $.notify("Massimo due lezioni al giorno", 'error');
                        }
                        calendar.refetchEvents();
                    }
                })
            }
        }else{
            $.notify("Ore non valide (Max. 4 ore e nell'intervallo 8-17)", 'error');
        }
    }else{
        $.notify("Nome lezione non valido (Min 3 caratteri)", 'error');
    }
}
}
```

Verifica delle ore:

```
function checkEvent(start, end){
    var s = new Date(start).setHours(0,0,0,0);
    var e = new Date(end).setHours(0,0,0,0);
    var st = new Date(start);
    var en = new Date(end);
    var hStart = st.getHours();
    var hEnd = en.getHours();
    var diffTime = en.getTime() - st.getTime();
    var diffHour = diffTime / (1000 * 60 * 60);
    if(s == e && diffHour <= 4 && hStart >= 8 && (hEnd < 17 || (hEnd == 17 && en.getMinutes() =
= 0))){
        return Math.ceil(diffHour);
    }else{
        return null;
    }
}
```

Verifica che le lezioni durino massimo 4 ore e che non siano prima delle 8 e dopo le 17.

Il metodo nel controller per l'aggiunta è il seguente:

```
public function insert(){
    require 'application/models/calendarioModel.php';
    $c = new CalendarioModel();
    $title = $this->test_input($_POST["title"]);
    $start = $this->test_input($_POST["start"]);
    $end = $this->test_input($_POST["end"]);
    $diff = $this->test_input($_POST["diff"]);
    return $c->insertEvents($title,$start,$end,$diff);
}
```

E il metodo nel model per l'aggiunta è il seguente:

```
public function insertEvents($title, $start, $end, $diff){
    $date = explode(" ", $start)[0];
    $selectDay = "SELECT * FROM giorno WHERE giorno='$date'";
    $day = $this->fetchAndExecute($selectDay);
    if($day != null) {
        $selectEvent = "SELECT * FROM lezione WHERE giorno='$date'";
        $event = $this->fetchAndExecute($selectEvent);
        if (count($event) < 2) {
            $query = "INSERT INTO lezione VALUES (null, '$title', '$start', '$end', '$date')";
            $statement = $this->connection->prepare($query);
            $statement->execute();
            $selectEvent = "SELECT * FROM lezione WHERE giorno='$date' and ora_inizio='$start'";
            $event = $this->fetchAndExecute($selectEvent);
            $id = $event[0]["id"];
            $query = "INSERT INTO assegna VALUES ('".$_SESSION["email"]."', '$id')";
            $statement = $this->connection->prepare($query);
            $statement->execute();
            $query = "UPDATE utente set ore_lavoro='".$$_SESSION["ore_lavoro"]-
$diff)."' where email='".$$_SESSION["email"]."'";
            $statement = $this->connection->prepare($query);
            $statement->execute();
            $_SESSION["ore_lavoro"] = $_SESSION["ore_lavoro"]-$diff;
            echo 1;
        }else{
            echo 3;
        }
    }else{
        echo 2;
    }
}
```

Per modificare la lunghezza di una lezione è sufficiente mettere il mouse alla fine della lezione e si potrà allungare:



Una volta “allungata” la lezione essa verrà aggiornata anche nel database.. Per fare ciò ho riempito il metodo “resize” della libreria del calendario che mi verificherà la ora di inizio e di fine lezione:

```
eventResize: function(eventResizeInfo) {
    var start = moment(eventResizeInfo.event.start).format("Y-MM-DD HH:mm:ss");
    var end = moment(eventResizeInfo.event.end).format("Y-MM-DD HH:mm:ss");
    var id = eventResizeInfo.event.id;
    if(checkEvent(start, end) != null) {
        $.ajax({
            url:"http://localhost/GestioneVacanze/Calendario/resize",
            type:"POST",
            data:{start:start, end:end, id:id},
            success: function () {
                $.notify("Lezione Aggiornata", "success");
                calendar.refetchEvents();
            }
        })
    }
    else{
        $.notify("Ore non valide (Max. 4 ore e nell'intervallo 8-17)", 'error');
        calendar.refetchEvents();
    }
}
```

Il metodo nel controller per modificare la lezione è il seguente:

```
public function resize(){
    require 'application/models/calendarioModel.php';
    $c = new CalendarioModel();
    $start = $this->test_input($_POST["start"]);
    $end = $this->test_input($_POST["end"]);
    $id = $this->test_input($_POST["id"]);
    return $c->resizeEvents($id,$start,$end);
}
```

Il metodo nel model per la modifica della lezione è il seguente:

```
public function resizeEvents($id, $start, $end){
    $date = explode(" ", $start)[0];
    $query = "UPDATE lezione SET ora_fine='$end' WHERE id=$id";
    $statement = $this->connection->prepare($query);
    $statement->execute();
    $selectEvent = "SELECT * FROM lezione WHERE giorno='$date' and ora_fine='$end'";
    $event = $this->fetchAndExecute($selectEvent);
    $id = $event[0]["id"];
    $hEnd = explode(":", $event[0]["ora_fine"])[0];
    $newHEnd = explode(":", explode(" ", $end)[1])[0];
    $diff = $newHEnd - $hEnd;
    $query = "UPDATE utente set ore_lavoro='".($SESSION["ore_lavoro"])."' where email='". $SESSION["email"]."'";
    $statement = $this->connection->prepare($query);
    $statement->execute();
    $SESSION["ore_lavoro"] = $SESSION["ore_lavoro"]-$diff;
}
```

Problemi riscontrati e soluzioni adottate

Durante la giornata di oggi non ho avuto particolari problemi.

Punto della situazione rispetto alla pianificazione

Sono in anticipo rispetto alla mia pianificazione di una settimana. Secondo il punto in cui mi trovo ora penso di concludere il progetto tra 2 settimane

Programma di massima per la prossima giornata di lavoro

La prossima giornata di lavoro inizierò la modifica delle lezioni tramite il calendario.