**Diario di lavoro -30.01.2017**

|  |  |
| --- | --- |
| Luogo | Trevano |
| Data | 30.01.2017 |

|  |
| --- |
| Lavori svolti |
| Il docente responsabile mi ha risposto tramite mail alla domanda di giovedì:   * Gli utenti registrati avranno tutti un numero svizzero?(07...)   *No direi che il numero potrebbe iniziare anche con altre cifre.*  Successivamente ho spostato il database da quello installato con workbench a quello presente nel pacchetto xampp. Riprovando a creare il sistema di login ho riscontrato un errore nella migrazione:  *SQLSTATE[42000]: Syntax error or access violation: 1071 Specified key was too long; max key length is 1000 bytes (SQL: alter table users add unique users\_email\_unique(email))*  Ho risolto il problema consultando il seguente forum: <https://laracasts.com/discuss/channels/general-discussion/syntax-error-or-access-violation-1071-specified-key-was-too-long> dove viene detto di aggiungere una riga di codice nel file *AppServiceProvider.php* nel metodo *boot*:  Schema::*defaultStringLength*(191);  Di base laravel crea una tabella *users* contenente l’id, il nome, l’email, la password e altri 3 campi che servono per recuperare la password. Per andare a modificare la struttura della tabella bisogna recarsi nel  File della creazione della tabella al percorso *database/migrations* e modificare la funzione *up()*:  public function up() {  Schema::*create*('users', function (Blueprint $table) {  $table->increments('id');  $table->integer('type');  $table->string ('name');  $table->string('surname');  $table->string('email')->unique();  $table->string('password');  $table->string('phone');  $table->dateTime('birthday');  $table->integer('state');  $table->rememberToken();  $table->timestamps();  }); }  Il metodo *create* della precedente funzione crea una tabella, dunque prima di iniziare la migrazione bisogna eliminare la tabella *users* esistente. Esiste anche la possibilità di agire in un’altra maniera, ovvero creare un nuovo file di migrazione e aggiungere le colonne mancanti ma ho scelto l’opzione di cancellare direttamente la tabella poichè più veloce e pratico.  Dunque dopo aver eliminato la tabella, effettuare una migrazione dai prompt dei comandi(*php artisan migration*).  In seguito ho creato la tabella dei progetti, per prima cosa andare nel prompt dei comandi e digitare:  *php artisan make:migration create\_project\_table*. Una volta eseguito il comando verrà creato il nuovo file di migrazione:  Aprire il nuovo file e ripetere le stesse operazioni fatte per creare la tabella *users* viste in precedenza.  Successivamente ho modificato la registrazione di base guardando la guida al seguente link: <http://www.easylaravelbook.com/blog/2015/09/25/adding-custom-fields-to-a-laravel-5-registration-form/>  Laravel inserisce solamente:   * Nome * Email * Password * Conferma della password   Ho aggiunto la parte dell’input nel file *register.blade.php*:  <div class="form-group{{ $errors->has('type') ? ' has-error' : '' }}">  <label for="type" class="col-md-4 control-label">Tipo</label>   <div class="col-md-6">  <input id="type" type="number" class="form-control" name="type" value="{{ old('type') }}" required autofocus>   @if ($errors->has('type'))  <span class="help-block">  <strong>{{ $errors->first('type') }}</strong>  </span>  @endif  </div> </div>  In seguito bisogna impostare che tipo di valore si vuole ottenere e dunque gestire, per fare ciò andare nel percorso *app/Http/controllers/auth* e modificare il file *RegisterController.php* aggiungendo le regole da gestire nella funzione *validator*:  protected function validator(array $data) {  return Validator::*make*($data, [  'email' => 'required|email|max:255|unique:users',  'password' => 'required|min:6|confirmed',  'type' => 'required|integer',  'name' => 'required|max:255',]); }  Sempre nella stessa pagina bisogna modificare la funzione *create*:  protected function create(array $data) {  return User::*create*([  'email' => $data['email'],  'password' => bcrypt($data['password']),  'type' => $data['type'],  'name' => $data['name'],]); }  Infine per inserire i dati nella query per creare l’utente, andare nel file *app/User.php* e modificare la variabile *$fillable*:  protected $fillable = [  'name', 'email', 'password','type' ]; |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Durante la migrazione dal database locale a quello di xampp ho riscontrato il seguente errore:  *SQLSTATE[42000]: Syntax error or access violation: 1071 Specified key was too long; max key length is 1000 bytes (SQL: alter table users add unique users\_email\_unique(email))*  Risolto andando a modificare il metodo *boot* nel file *AppServiceProvider.php* e aggiungendo il seguente codice:  Schema::*defaultStringLength*(191);  Soluzione trovata dal seguente forum: <https://laracasts.com/discuss/channels/general-discussion/syntax-error-or-access-violation-1071-specified-key-was-too-long> |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| In linea con la pianificazione |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Finire definitivamente l’analisi e la progettazione e riprotare il tutto sulla documentazione. |