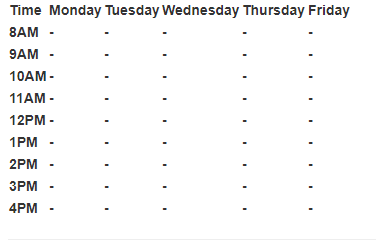
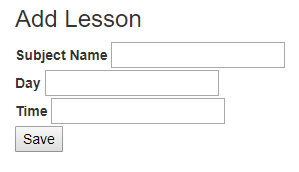
Timetable Anleitung

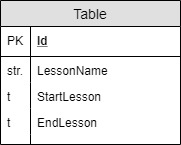
# Vorbereitung

1. Ich habe ein Repository auf GitHub erstellt, welches «Timetable» heisst.
2. Danach habe ich eine neue ASP.NET Solution erstellt. Ich habe dieses Solution in das geklonte Timetable Repository gespeichert.
3. Danach habe ich auf gitignore.io ein File generieren lassen, welches die Files, welche nicht versioniert werden müssen bei jedem Commit ignoriert werden. Dieses speichere ich dann mit dem Filenamen: .gitignore
4. Ich schaue mir die Solution an. Normalerweise erstellt Visual Studio schon ein gutes Grundgerüst. Je nachdem was man davon braucht, kann man schon einige Views etc. löschen. Wenn man jedoch alles braucht, kann man es auch einfach lassen.
5. Wenn man mit dem Grundgerüst zufrieden ist, muss man die benötigten Files auf das Git Repository laden. Dabei muss man achten, das alle nötigen Files für die Applikation mit Tortoise Git via Add hinzugefügt wurden.  
   Folgende Files/Ordner sollte man dabei nicht auf das Repository laden:
   1. .vs
   2. .git
   3. Alle .dll Files
   4. bin
   5. obj
   6. fonts
6. Wenn alles Commited ist, und ich mir sicher bin, das alle Files die ich brauche da sind, kann ich das ganze Projekt pushen. Vor dem Push habe ich im Repository ein Word Dokument erstellt, welches auch für dieses Projek benötigt wird.

# Implementation

1. Bei der Implementation von ASP.NET Applikationen, verwende ich immer die gleiche Routine. Dabei ist der Fokus zuerst auf das grobe Design des User Interfaces. Es muss noch nicht gut sein, jedoch sollten alles Views und deren Inhalt schon da sein. Wenn es Tabellen oder sonstiges gibt, platziere ich vorerst Mockdaten in diese.  
   In diesem Projekt habe ich zwei Views. Eine Timetable und eine Subject View. Bei der Timetable ist der Stundenplan ersichtlich. Von Montag bis Freitag kann man jeden Tag die Stunden von 8AM bis 4PM eintragen.  
   

Bei der Subject View kann man neue Lektionen verfassen, und denen eine Zeit geben. Wenn man eine Lektion speichert, sollte sie auf dem Stundenplan ersichtlich sein.   
Bei der View müssen drei Inputs sein, und zwar der Name der Lektion, der Tag und die Zeit. Zudem braucht es einen Save Button, um die Lektion zu speichern.  


1. Wenn die Views erstellt worden sind, erstelle ich die Datenbank dafür. Dies mache ich im Visual Studio mithilfe vom Designer. Auf folgender Abbildung sieht man das ERD der Datenbank.  
     
   Jede Lesson sollte einen Namen haben, und eine Start- und Endzeit. Diese Zeit sollte dann auf der Tabelle angezeigt werden.