Path Planning Project:

1. The car is able to drive at least 4.32 miles without incident:

   The current/best miles driven without incident is verified up to 12 miles. The car could successfully navigate 12 miles without any incidence i.e. car maintained the max limits for acceleration (10 m/s^2), jerk (1 m/s^3) and speed (50 mph). There were no collisions on the same lane and during lane change. Car was driven inside of the lanes at all the times except for lane change within 3 seconds. List of widely spaced (x,y) waypoints is created which is evenly spaced at 30m. These waypoints are interpolated with Spline. “Spline.h” is included in “main.cpp”.

2. The car drives according to the speed limit:
   The car doesn't drive faster than the speed limit of 50 mph. Also, the car isn't driving much slower than speed limit unless obstructed by traffic.
   The variable MAX_SPEED is defined at line #214. This max speed is defined for value 49.70 miles per hour. If car reaches this speed, the break is applied by reducing the speed by 0.5mph.
   The logic at line #499 to #514 defines the way to regulate the speed of car. If current car speed reaches max speed then it starts reducing the speed by 0.5mph (0.224 meters per second) in every iteration. Else if speed is not reaching the max speed limit, it increases the speed by 0.5mph (0.224 meters per second) in every iteration.

3. Max Acceleration and Jerk are not Exceeded and Car does not have collisions:
   The variable defined at line # 215 is DIST_TOO_CLOSE_BREAK. So, if car in front is at distance 30 miles or less then break is applied to avoid collision.
   The logic at line #499 to #514 defines the way to regulate the speed of car to avoid collusion. Below is the code snippet. If code detects that car in front is closer than 30 miles, then it turns the accident_possible flag on and speed is reduced by 0.5mph (0.224 meters per second) in every iteration. The car does not exceed a total acceleration of 10 m/s^2 and a jerk of 1 m/s^3.

```
if (accident_possible)
{
	current_car_speed = current_car_speed - 0.224
}
else if (current_car_speed < MAX_SPEED)
{
	if (current_car_speed < MAX_SPEED-10)
	{
		current_car_speed = current_car_speed + 0.224*1.5
}
	else
	{
		current_car_speed = current_car_speed + 0.224;
	}

}
```

4. The car stays in its lane, except for the time between changing lanes.
   The car doesn't spend more than a 3 second length outside the lane lanes during changing lanes, and every other time the car stays inside one of the 3 lanes on the right hand side of the road.

5. The car is able to change lanes
   Lane variable drives the logic of changing the lanes. It is initiated at line #204. The variable is initiated to 1 which is center lane.
   DIST_TOO_CLOSE_CHANGE_PATH variable is initialized to 40 miles. If car in front is in 40 miles range then logic is written to check the cars in adjacent lanes. If car in the adjacent lane is at 40 miles distance in front and/or car in the adjacent lane is at 5 miles distance in back then it is determined to be safe to change the lane based on the speeds of the cars. These variables are defined at the line # 217 and 218.
   Code from line #273 to #491 defines the logic for lane change. If there is a car in the same lane as of our car then check the speed of the other car. If other car's future ‘s’ value is greater than our car's future ‘s’ value and distance between them is less than 30m then take action to change the lane. Depending on, in which lane the car is, the logic is written to check if lane change is safe based on cars in front and cars in the back. If lane change is determined to be safe then lane is changed by updating the ‘lane’ variable value.