

## به نام خدا

تشریح پروژه پایانترم درس ساختمان‌های داده

دانشگاه هرمزگان ترم ۴۰۰۱ - دکتر شهرام گلزاری

تاریخ شروع:

آخرین مهلت تحویل:

## فهرست موضوعی

2	مقدمه
3	دوربین‌های نظارتی
4	خودروها
5	پلاک
5	کنترل هوشمند سرعت
7	خودروهای سرقتی
7	تخلف‌ها
8	فایل تست ورودی
9	فایل خروجی
9	گیت‌ها و پیاده‌سازی
10	پنل کاربری

## مقدمه

مدیریت حمل و نقل استان هرمزگان به تازگی تعدادی دوربین نظارتی جهت مدیریت وضعیت حمل و نقل در سطح استان خریداری و نصب کرده است، اکنون به یک نرم‌افزار جهت اتوماسیون برخی امور نیاز است.

به صورت کلی خودروها از چندین جهت ممکن است مرتکب تخلف شوند، نرم‌افزار می‌بایستی تمامی این شرایط را برای هر خودروی عبوری از هر کدام از دوربین‌های نظارتی بررسی نماید، بدیهی است در هر واحد زمانی تعداد زیادی خودرو از این دوربین‌ها عبور می‌کنند و مهمترین مسأله در این زمینه هزینه زمانی بالا یا نیاز به کامپیوتری بسیار قدرتمند برای برنامه است، اما پس از مشورت با متخصصین در این حوزه، شما به عنوان یک متخصص طراحی الگوریتم و ساختمان داده معرفی شده‌اید تا راهکاری با حداقل نیاز به منابع کامپیوتری (که مهمترین آن‌ها پردازنده مرکزی و حافظه اصلی کامپیوتر است) ارائه کنید.

هر دوربین نظارتی می‌تواند در محدوده‌ی خارج از شهر یا داخل شهر باشد، دوربین‌های داخل شهر باید صرفاً در ساعات خاصی اجازه ورود به خودروهای سنگین و باری به درون شهر را بدهند، اما این محدودیت برای دوربین‌های خارج از شهر وجود ندارد، هر دوربین با توجه به محل نصب، حداکثر سرعت مجاز برای خودروهای سواری و سنگین (در دو گروه) در نظر می‌گیرد، همچنین برای برخی دوربین‌ها ممکن است مدیر سیستم حداقل سرعت مجاز نیز تعریف کند، علاوه بر آن برخی خودروها به عنوان خودرو سرقتی در سیستم تعریف شده‌اند، بلافاصله پس از عبور یک خودروی سرقتی برنامه باید به مدیر سیستم گزارش مربوطه را ارائه کند.

هرچند تمامی تخلف‌ها در یک منو از برنامه گزارش می‌شود اما گزارش خودروی تخلفی باید در بالای لیست گزارش‌ها (ترجیحاً به رنگ قرمز برای جلب توجه مدیر سیستم) قرار بگیرد تا احتمال عدم توجه مدیر سیستم کاهش یابد.

برنامه شامل دو بخش اصلی دوربین‌ها و خودروها می‌باشد، در بخش دوربین‌ها کاربر می‌تواند لیستی از دوربین‌ها را مشاهده کند، برای هر کدام تنظیمات سرعت و دیگر اطلاعات را تغییر دهد، یا اینکه دوربین را خاموش کند، همچنین می‌تواند دوربین جدیدی در سیستم تعریف کند. در بخش خودروها علاوه بر مشاهده لیستی از خودروها، تغییر دادن هر کدام از فیلدها و افزودن خودروی جدید، مدیر سیستم می‌تواند خودرویی را به عنوان سرقتی علامت بزند.

36 یک قابلیت مهمی که برنامه باید به آن مجهز شود، سیستم کنترل سرعت هوشمند با  
استفاده از چند دوربین است، فرض کنید دو دوربین در ابتدا و انتهای یک جاده قرار دارند،  
38 فاصله این دو دوربین به کیلومتر مشخص، و حداکثر سرعت مجاز نیز مشخص می‌باشد،  
40 اکنون اگر خودرویی سریعتر از حداقل زمان ممکن به دوربین دوم برسد، این خودرو مرتکب  
تخلف سرعت غیر مجاز شده و بایستی جریمه شود.

در ادامه به تفصیل بخش‌ها و عملکردهای برنامه ارائه می‌شود.

## 42 دوربین‌های نظارتی

بدیهی است که در ابتدای شروع به کار نرم‌افزار باید تمامی دوربین‌های موجود ابتدا وارد  
44 سیستم شوند و سپس نرم‌افزار شروع به کار کند. برای هر دوربین ورودی‌های زیر از کاربر  
دریافت می‌شود.

46 ۱. نام دوربین: یک نام به صورت یک رشته (استرینگ) با حداکثر طول ۴۰ حرف، این  
نام بعداً در برنامه جهت جست‌وجو در لیست دوربین‌ها مورد استفاده قرار می‌گیرد،  
48 دقت شود که نام یکتا نیست و دو دوربین می‌توانند نام یکسانی داشته باشند.

۲. آدرس: آدرس محل نصب دوربین به صورت یک رشته (استرینگ) با حداکثر طول  
50 ۲۰۰ حرف از کاربر دریافت می‌شود، این فیلد اختیاری است و کاربر ممکن است برای  
دوربین آدرسی در نظر نگیرد.

52 ۳. کد: یک کد چهار رقمی که یک عدد از ۱۰۰۰ تا ۹۹۹۹ است، این کد یکتا می‌باشد و  
هر دوربین کد مربوط به خود را دارد. حداکثر تعداد دوربین‌هایی که برنامه پشتیبانی  
54 می‌کند ۹۰۰۰ تا است. هرچند تاکنون حدود ۲۰۰۰ دوربین خریداری و نصب  
گردیده است و پیشبینی می‌شود تا چند سال آینده نیازی به افزایش این تعداد  
56 نیست.

۴. خارج/داخل منطقه شهری بودن دوربین، در این مورد یک گزینه دو حالتی وجود دارد  
58 که کاربر تعیین می‌کند این دوربین خارج یا داخل شهر قرار دارد. پیشتر اشاره شد  
که دوربین‌های داخل شهر در زمان‌های غیر مجاز کامیون‌ها و خودروهای سنگین را  
60 جریمه می‌کنند.

۵. حداکثر سرعت مجاز برای کامیون، و برای خودرو سواری. (این نوع جریمه برای هر  
62 دوربین ممکن است غیرفعال باشد، این فیلد اختیاری است)

۶. حداقل سرعت مجاز عبور از این جاده، نوع خودرو در این فیلد اطلاعاتی اهمیتی  
64 ندارد. (اختیاری)

فاصله این دوربین از دوربین‌های مجاور برای کنترل هوشمند سرعت مجاز در بخش دیگری از برنامه تعیین می‌شود، و در لحظه تعریف دوربین جدید این اطلاعات از کاربر دریافت نمی‌شود.

## خودروها

خودروها به طور کلی در دو دسته سواری و سنگین تعریف می‌شوند. تعداد خودروها در برنامه بسیار زیاد، در لحظه راه اندازی ابتدایی حدود ۵۰۰ هزار خودرو باید تعریف شوند، این تعداد هر ساله حدوداً ۵۰ هزار تا افزایش می‌یابد (البته این مقدار تعیین شده و دقیق نیست، برنامه نباید در این زمینه محدودیتی داشته باشد).

برای هر خودرو فیلدهای زیر را در برنامه ذخیره کنید:

۱. مدل خودرو: یک گروه مدل خودرو در نرم‌افزار تعریف شده، مدل خودرو باید یک مقدار از بین این گروه باشد، اکنون به دلیل بالا رفتن هزینه پروژه کارفرما صرفاً نام نوع خودرو را مد نظر دارد، اما خواسته تا امکان افزودن اطلاعاتی اختصاصاً برای یک مدل خودرو وجود داشته باشد، این امکان باعث می‌شود در توسعه‌های بعدی، نرم‌افزار بتواند دوره معاینه فنی، تعویض روغن و.. هر خودرو با توجه به مدل آن بررسی کند.

۲. نام مالک خودرو: این فیلد صرفاً یک رشته (استرینگ) است.

۳. کد ملی مالک خودرو: این فیلد صرفاً یک رشته (استرینگ) ده رقمی است، که همه کارکترهای آن عددی هستند.

۴. سنگین یا سبک بودن خودرو: این فیلد اطلاعاتی برای بررسی حداکثر سرعت مجاز در برخی دوربین‌ها که بسته به نوع خودرو متفاوت است استفاده می‌شود، همچنین دوربین‌های سطح شهر با توجه به این فیلد خودروهای سنگین را در ساعات خاصی از شبانه‌روز جریمه می‌کنند. بدیهی است دوربین‌هایی که بررسی کردن وضعیت جریمه یک خودرو در آن‌ها وابسته به این فیلد نیست (مثلاً دوربینی که در خارج شهر تعریف شده و فقط حداکثر سرعت ۸۰ تعریف شده) جهت کاهش هزینه‌های زمانی نیازی به بررسی این فیلد برای هر خودروی عبوری نیستند.

۵. پلاک خودرو: در اینجا صرفاً یک رشته (استرینگ) از کاربر دریافت کنید، جهت نحوه بررسی صحت پلاک و الگو یک پلاک یک تیترا مجزا ارائه شده.

هر پلاک خودرو شامل یک عدد دو رقمی، یک حرف، یک عدد سه رقمی و یک عدد دو رقمی دیگر است. به طول کل اگر تمامی حالات را در نظر بگیریم می‌توان ۳۲۰ میلیون خودرو پلاک کرد، از آنجایی که در بخش خودروها مطرح شد که تعداد خودروها حدوداً ۵۰۰ هزار و با روندی نسبتاً ملایم رو به افزایش است. پلاک دریافتی از هر ورودی برنامه ممکن است متفاوت باشد، مثلاً پلاک دریافتی از ورودی فایل تست (جلوتر ارائه می‌شود) به صورت

25-2-325-84

است، در پلاک فوق عدد دو در قسمت دوم پلاک همان حرف است جهت استفاده از صرفاً اعداد، شماره کارکتر مذکور در نظر گرفته می‌شود، در مثال فوق با در نظر گرفتن عدد صفر برای حرف الف و یک برای ب، عدد دو برای حرف پ خواهد بود. باقی ارقام عیناً همان اعداد روی پلاک هستند.

در هنگام دریافت ورودی پلاک از کاربر می‌توانید حرف پلاک را به صورت یک لیست چند گزینه‌ای از کاربر دریافت کنید (tkinter در combo box)، اما بدیهی است که دریافت آن به صورت عدد همانند حالت قبل باعث اخراج شدن شما از شرکت مذکور می‌شود.

بهتر است برای نمایش پلاک در منوهای کاربری از حرف فارسی مربوطه استفاده کنید.

یک راهکار مناسب در تمام برنامه تبدیل هر کدام از حالات ورودی/خروجی پلاک به یک حالت استاندارد (مثلاً یک عدد) است، تا در تمام کد از همان حالت استاندارد بهره ببرید.

## کنترل هوشمند سرعت

مدیر سیستم می‌تواند با مراجعه به یک پنل در برنامه، سیستم کنترل هوشمند بین دو دوربین نظارتی را فعال کند، هرچند برای هر دوربین می‌توان به طور همزمان با چندین دوربین که از جاده‌های متفاوتی به یک دیگر متصل هستند، سیستم کنترل هوشمند سرعت را فعال نمود، اما هر نسخه از کنترل هوشمند صرفاً بین دو دوربین که ابتدا و انتها هستند تنظیم می‌شود.

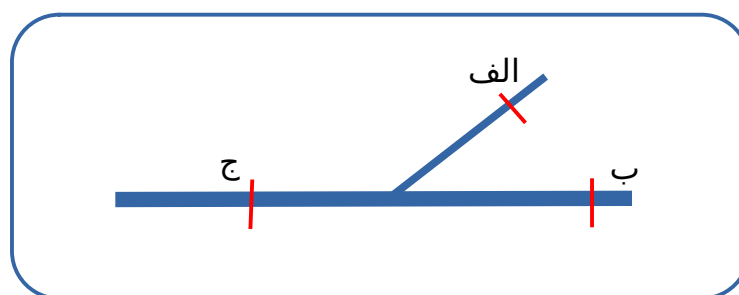
برای دو دوربینی که در ابتدا و انتهای یک مسیر قرار دارند، با در نظر گرفتن حداکثر سرعت مجاز و فاصله دوربین‌ها می‌توان حداقل زمان رسیدن به دوربین انتهایی بازه پس از حرکت از دوربین ابتدایی با حداکثر سرعت مجاز را محاسبه نمود، فرض کنید حداکثر سرعت مجاز ۱۲۰ کیلومتر بر ساعت، و فاصله دو دوربین در یک محور ۶۰ کیلومتر است، حداقل زمان حاصل تقسیم ۶۰ بر ۱۲۰ است، یعنی نیم ساعت.

برنامه شما می تواند از کاربر حداقل زمان را دریافت کند، و یا اینکه فاصله و حداکثر سرعت مجاز را دریافت و حداقل زمان را محاسبه کند. بهر حال پس از مشخص شدن حداقل زمان برنامه باید خودروهایی که در زمانی کمتر از زمان فوق مسیر را طی می کنند جریمه کند.

بدیهی است که اگر خودرویی در زمان صفر در دوربین اول مشاهده، و نیم ساعت بعد همچنان به دوربین دوم نرسیده بود نیازی به بررسی این خودرو از جهت کنترل هوشمند سرعت دیگر نیست، با این تکنیک می توان تعداد خودروهای مورد بررسی در دوربین دوم را کاهش داد.

البته از آنجایی که مسیرهای بین دو دوربین کاملاً بسته نیستند، پس ممکن است تعدادی خودرو در دوربین دوم مشاهده شوند که در دوربین اول مشاهده نشده اند، این خودروها از مسیرهای فرعی بعد از دوربین اول وارد مسیر اصلی شده اند. در پیاده سازی الگوریتم و ساختارهای این بخش برنامه باید این مورد در نظر گرفته شود.

کنترل هوشمند سرعت ممکن است بین چندین دوربین به صورت مشترک فعال باشد، فرض کنید دو مسیر به صورت زیر وجود داشته باشند:



دوربین های الف، ب و ج به صورت فوق واقع شده اند، در این مثال مدیر سیستم برای کنترل هوشمند سرعت باید دو بار کنترل هوشمند سرعت را یک بار برای محور الف-ج و یک بار برای محور ب-ج تنظیم کند، در نتیجه در دوربین ج باید خودروهای ورودی از الف و ب مورد بررسی قرار بگیرند.

مثال فوق می تواند بلعکس نیز باشد، یعنی ج ورودی، الف و ب خروجی باشند، در این صورت مدیر سیستم دو بار کنترل هوشمند سرعت برای محور ج-الف و ج-ب تنظیم می کند.

در نتیجه باتوجه به موارد فوق برای هر دوربین ورودی، یک تعداد دوربین خروجی وجود دارد که ممکن است ماشین ها از آنجا خارج شوند و برای هر دوربین خروجی تعدادی دوربین ورودی وجود دارد که ممکن است ماشین ها از آنجا وارد شده باشند. با این شرایط، الگوریتم و ساختاری بایستی ارائه شود که در اولویت اول از جهت زمانی منطقی و در اولویت دوم از جهت مکانی حافظه زیادی اشغال نکند.

## خودروهای سرقتی

148 در برنامه و سطح استان تعدادی خودرو به عنوان خودرو سرقتی مشخص شده اند، این  
خودروها به محض مشاهده در یکی از دوربین ها برنامه باید فوراً ادمین سیستم را مطلع کند.  
150 تعداد این خودروها حدوداً بین ۱۰۰ تا ۵۰۰ خودرو در فصل های سال تغییر می کند،  
ادمین سیستم به صورت دستی پس از گذشت یک سال از پیدا نشدن، یا پیدا شدن خودرو  
152 آن را از وضعیت سرقتی خارج می کند، پس برای هر خودرو قابلیت مشخص کردن وضعیت  
سرقتی یا عدم این وضعیت باید در اختیار باشد.  
154 از آنجایی که تعداد خودروهای سرقتی همواره بسیار کمتر از کل تعداد خودروها است،  
بسته به نوع پیاده سازی ممکن است بهتر باشد برای هر خودرو مشاهده شده صرفاً  
156 خودروهای سرقتی بررسی و نیازی به جست و جو در کل خودروها نباشد.

## تخلف ها

158 هر خودروی عبوری از یک دوربین ممکن تخلف های زیر را مرتکب شده باشد، هر تخلف  
با شماره آن، که از یک شروع می شود شناخته می شود.  
160 ۱. سرعت زیاد غیر مجاز: خودرویی که با سرعتی بالاتر از سرعت مشخص شده از یک  
دوربین عبور کند این تخلف را مرتکب شده است. (ممکن است برای یک دوربین  
162 هیچ حداکثر سرعت مجازی تعریف نشده باشد، در این صورت این تخلف در این  
دوربین بررسی نمی شود)  
164 ۲. سرعت پایین غیر مجاز: خودرویی که پایین تر از سرعت مجاز تعیین شده برای یک  
دوربین از آن عبور کرده باشد این تخلف را مرتکب شده است. (این تخلف نیز همانند  
166 مورد قبل در صورت عدم تعریف در دوربین بررسی نمی شود)  
168 ۳. کنترل هوشمند سرعت: در این تخلف در صورتی که خودرو زودتر از حداقل زمان  
ممکن به دوربین پایان بازه از دو دوربینی که کنترل هوشمند برای آن ها تنظیم شده  
برسد مرتکب این تخلف شده است.  
170 ۴. سرقتی: این نوع تخلف در همه دوربین ها بررسی می شود، در صورتی که خودرویی به  
عنوان سرقتی علامت گذاری شده باشد و در یکی از دوربین ها مشاهده شود این  
172 تخلف گزارش می شود.

## فایل تست ورودی

از آنجایی که برنامه شما در حالت آزمایشی قرار دارد و بعداً باید به نسخه نهایی تبدیل شود، برای تست کردن برنامه یک فایل در پوشه نرم افزار قرار داده می شود که به آن فایل تست می گوییم، برای شبیه سازی در مدل واقعی این فایل مشخص می کند در چه زمانی چه خودروهایی در چه دوربینی مشاهده شده اند. با وجود این برنامه شما پس از شروع باید این فایل را باز کند و با خواندن اطلاعات درون آن شبیه سازی مدل واقعی را آغاز کند.

هر خط از فایل مشخص کننده یک ثانیه در مدل واقعی است، در این خط لیستی از تمامی ماشین های مشاهده شده قرار دارد، برنامه در هر ثانیه باید یک خط از این فایل را بخواند (می توانید برای سرعت بیشتر هر ده ثانیه یک بار ده خط را بخوانید، این مسأله در درس ساختمان های داده مد نظر نیست) سپس بررسی های لازم را انجام داده و در صورت بروز تخلفی آن را گزارش کند، در نهایت یک ثانیه صبر کند و خط بعدی را بخواند.

در فایل تستی که در اختیار شما قرار می گیرد برنامه به مدت ۱۲۰ ثانیه مورد تست قرار می گیرد، در نتیجه این فایل ۱۲۰ خط دارد، در هر خط داده ها با کارکتر کاما (,) از یک دیگر جدا شده اند، اولین داده درون یک خط شماره آن خط است، داده های بعد از آن (در صورت وجود) ماشین های مشاهده شده در دوربین ها است، و هر کدام از داده ها به صورت کد چهار رقمی دوربین سپس یک دو نقطه (: ) و کد پلاک ماشین مشاهده شده و بعد از یک دو نقطه دیگر سرعت ماشین تعریف شده اند.

برای مثال فایل زیر به عنوان فایل تست در نظر بگیرید:

0,5326:24-0-356-84:45,3695:26-1-222-84:65

1,5326:25-0-659-84:120

2

در فایل فوق در ثانیه صفر یک خودرو با پلاک ۲۴ الف ۳۵۶ و ۸۴ از دوربین با آیدی ۵۳۲۶ عبور کرده که سرعت آن ۴۵ کیلومتر بر ساعت بوده و یک خودرو دیگر با پلاک ۲۶ ب ۲۲۲ و ۸۴ از دوربین با آیدی ۳۶۹۵ با سرعت ۶۵ کیلومتر بر ساعت عبور کرده است، به همین صورت ثانیه یک، در ثانیه دو هیچ خودرویی عبور نکرده.

ابتدا در برنامه شما لیستی از خودروها و دوربین ها به صورت پیش فرض برای تست تعریف می شوند، سپس طبق فایل تست شروع به شبیه سازی می شود. خروجی برنامه در بخش بعدی مطرح می شود.

لیست خودروها، دوربین ها و تنظیمات اولیه برنامه برای شروع تست به همراه فایل تست در یک فایل خوانا و مجزا مشخص می شود، البته از آنجایی که تمامی اطلاعات برنامه باید ذخیره شوند تا در دفعات بعدی که برنامه اجرا می شود داده ها از قبل وجود داشته باشند



یک بار کافی است داده‌های درون فایل تنظیمات اولیه را وارد برنامه کنید و سپس شبیه سازی برنامه را به صورت مکرر انجام دهید.

## فایل خروجی

اگرچه در پنل کاربری خروجی برنامه قابل مشاهده است، که جلوتر بخش‌های مورد نیاز ارائه می‌شود، اما در یک فایل باید در حین شبیه سازی تا پایان یافتن فایل تست گزارش تخلفات در یک فایل نوشته شود، اینکار بررسی عمل کرد صحیح برنامه را آسان تر می‌کند.

فرمت فایل خروجی شبیه به فایل تست است، در این فایل که یک فایل متنی است به ازای هر ثانیه از صفر تا  $n$  ثانیه شبیه سازی (که در فایل تست ۱۲۰ ثانیه تعریف شده) یک خط در این فایل می‌نویسد که داده‌های آن با استفاده از کاما (,) از یک دیگر جدا شده اند، داده اول درون هر خط شماره ثانیه و داده‌های بعد از آن (در صورت وجود) تخلف‌های ثبت شده هستند، هر داده شامل دو دو نقطه (:) است که عنصر اول آیدی دوربین، عنصر دوم پلاک خودرو و عنصر آخر کد نوع تخلف است (یک عدد از یک تا چهار).

به عنوان مثال در صورتی که این فایل به صورت زیر باشد:

```
0
1,5326:25-0-659-84:1
2
```

در ثانیه‌های صفر و دو هیچ تخلفی صورت نگرفته، و در ثانیه یک دوربین ۵۳۲۶ خودرویی با پلاک ۲۵ الف ۶۵۹ و ۸۴ را با سرعت زیاد غیر مجاز مشاهده کرده.

پس از اجرای برنامه باید فایلی به نام output.txt در همان پوشه پروژه ایجاد شده باشد (یا رونویسی شود) که در آن با فرمت فوق به تعداد ثانیه‌های تعریف شده در فایل تست (test.txt) خط وجود دارد.

بدیهی است برای یک فایل تست و تنظیمات اولیه برنامه مشخص، خروجی دقیقاً یکی و مشخص می‌شود، لذا در وهله اول کار کرد صحیح برنامه شما بررسی و سپس الگوریتم‌ها، ساختارها و ساختمان داده‌های مورد استفاده بررسی می‌شوند.

## گیت‌هاب و پیاده‌سازی

جهت امنیت و سادگی کار انجام دهنده پروژه و مدیریت بهتر و تسلط کامل به پروژه توسط تصحیح کننده، برای هر دانشجوی درس یک رپوزیتوری در گیت‌هاب توسط حل التمرین ایجاد می‌شود، سپس انتظار می‌رود تمامی توضیحات پروژه در فایل README.md

نوشته شود، این توضیحات باید تا توضیح دادن عمل کرد، ورودی‌ها و خروجی‌های تمامی  
توابع و کلاس‌های تعریف شده در برنامه جزئی باشد، البته رابط کاربری یا کدهای مربوط به  
GUI نیازی به توضیحات ندارد و یا اینکه مختصر و مفید ارائه شود.

کامیت (commit) های متعدد و پوش (push) های روزانه یا ساعتی لازم است، این امر  
باعث می‌شود تا روند انجام پروژه قابل پیگیری توسط حل التمرین و استاد درس باشد، از  
طرف دیگر در صورتی که فایل یا کدی به هر دلیلی از روی کامپیوتر شخصی حذف یا اینکه  
به هر دلیلی دسترسی دانشجو به کامپیوتر شخصی وجود نداشته باشد، کد در محلی  
محفوظ ذخیره شده و قابل بازیابی است.

در صورتی که دانشجو قصد انجام یک آزمایش روی کد خود داشته باشد و می‌خواهد  
نسخه پایدار کد را در محلی جدا ذخیره کند می‌تواند از برنج (branch) استفاده کند، این  
امر باعث می‌شود تا در زمان تحویل و ارائه پروژه در صورتی که از دانشجو خواسته شود  
تغییراتی در کد خود اعمال کند، ابتدا یک برنج ایجاد و سپس تغییرات در آن برنج انجام  
شود، تا از شکسته (break) شدن کد اصلی جلوگیری شود.

پس از ایجاد رپوزیتوری دانشجویان یک ایمیل از گیت‌هاب مبنی بر دعوت شدن به  
رپوزیتوری دریافت می‌کنند، دانشجو پس از باز کردن این لینک باید دعوت به رپوزیتوری را  
بپذیرد (accept invitation) و پس از کلون (clone) کردن شروع به انجام پروژه کند.

## پنل کاربری

اگرچه تمامی بخش‌های مورد نیاز نرم‌افزار مطرح شده اما در این بخش به صورت مجزا  
اعضاء تشکیل دهنده پنل یا رابط کاربری برنامه شما مشخص می‌شود.

برای این نرم‌افزار یک پنل کاربری ایجاد کنید به طوری که پس از اجرای برنامه،  
مستقیماً این پنل باز شود (نیازی به ورود کاربر وجود ندارد)، در این برنامه تنها یک کاربر  
وجود دارد و آن هم مدیر سیستم است.

در این پنل به صورت کاربر پسندی بخش‌های زیر را در دسترس قرار دهید:  
۱. دوربین‌ها:

۱. افزودن یک دوربین جدید: با دریافت فیلدهایی که پیشتر مشخص شد یک  
دوربین جدید به لیست دوربین‌ها اضافه می‌شود.

۲. مشاهده لیست دوربین‌ها: لیستی از دوربین‌ها به صورتی که برای هر دوربین نام،  
کد، حداکثر سرعت مجاز (در صورتی که تعریف شده باشد) در یک ستون باشد  
نمایش دهید.

۳. افزودن کنترل هوشمند سرعت: با انتخاب این گزینه دو دوربین انتها و ابتدای بازه و یک حداقل زمان به ثانیه از کاربر دریافت می‌شود، سپس کنترل سرعت بین این دو دوربین فعال می‌شود.

## ۲. خودروها:

۱. افزودن یک خودرو جدید: با انتخاب این گزینه تمامی اطلاعات مربوط به یک خودرو که در بخش‌های قبل مشخص شد از کاربر دریافت می‌شود و پس از تأیید خودرو در برنامه اضافه می‌شود.

۲. مشاهده لیست خودروها: این بخش شامل یک لیست از خودروهای تعریف شده در برنامه می‌باشد، دقت کنید نمایش دادن و بارگزاری تمامی خودروها یک‌جا عملیاتی منطقی نمی‌باشد (به تعداد خودروها توجه کنید)، برای هر خودرو نام مالک، کد ملی مالک، شماره پلاک و نوع آن را در ستون‌ها نمایش دهید.

۳. کادر جست و جو: در این کادر کاربر می‌تواند با تایپ کردن **پلاک خودرو** یا **کد ملی مالک** یا **نام مالک** در میان خودروها جست‌وجو انجام دهد. نتایج جست‌وجو (تمامی مطابقت‌ها) باید به صورت لیستی که در مورد قبل گفته شد نمایش داده شود.

۴. علامت گذاری خودرو به عنوان سرقتی: برای یک خودرو مدیر سیستم می‌تواند با انتخاب این گزینه، خودرو را به عنوان سرقتی مشخص کند، از این لحظه در صورتی که خودرو در یکی از دوربین‌ها مشاهده شود باید گزارش مربوطه توسط نرم‌افزار ارائه شود.

۵. علامت برداری سرقتی: با انتخاب یک خودرو که سرقتی است می‌توان علامت سرقتی بودن را از روی آن برداشت.

۶. مشاهده لیست تخلفات: برای یک خودرو انتخابی می‌توان لیست تخلفاتی که تاکنون مرتکب شده را مشاهده کرد. برای هر تخلف صرفاً نام آن ذکر شود کافی است.

## ۳. کنسول:

در این بخش کاربر می‌تواند تخلفات لحظه‌ای (برای همه خودروها و همه دوربین‌ها) را مشاهده کند. برای هر تخلف نام دوربین، شماره پلاک خودرو و نوع تخلف را مشخص کنید.

موفق باشید.