



دانشکده فنی مهندسی - گروه کامپیوتر

پایان نامه کارشناسی

سفرنامه

نگارش:

مهدی یاسینی

استاد راهنما:

دکتر طباطبائی

بهمن ۱۳۹۹



## چکیده

امروزه تقریباً تمامی تصمیمات و رخداد‌های روزانه جامعه بشری اعم از تجارت، خرید، پزشکی و... به صورت مستقیم و غیر مستقیم با دنیای دیجیتال و به طور دقیق‌تر اینترنت پیوند خورده است؛ بنابراین ورود موضوع گردشگری و سفر با تمامی حوزه‌های اصلی و فرعی آن به این دنیای ناپیدا چندان به دور از پیش بینی نبود و طبق گزارش‌های سالانه، موضوع گردشگری را به عنوان یکی از سودآورترین حوزه‌های اینترنتی در ایران نام برده که مهر تاییدی بر اهمیت این حوزه است.

علی بابا، مستر بلیت، جاباما (زیر مجموعه گروه علی بابا) را می‌توان به عنوان مهم‌ترین بازیگران این حوزه نام برد.

اما موضوعی که از آن غفلت شده، "سفر" است که در حاشیه حوزه‌های پر رنگ‌تر این حوزه مثل بلیت فروشی و رزرو هتل قرار گرفته است.

**کلمات کلیدی: سفرنامه-وبلاگ-سفرنویسی-وب سایت**

## فهرست

۸.....	مقدمه
۹.....	فصل اول ابزارهای مورد استفاده.....
۱۰.....	محیط برنامه نویسی
۱۱.....	زبان برنامه نویسی
۱۱.....	کجا استفاده می شود؟
۱۳.....	بخش سرور
۱۳.....	فریم ورک (Framework)
۱۳.....	نود جی اس (Node Js)
۱۷.....	مونگو دی بی (Mongo DB)
۱۸.....	کلاینت
۱۸.....	HTML
۱۹.....	CSS
۱۹.....	JQuery
۱۹.....	Bootstrap
۱۹.....	EJS
۲۰.....	MVC
۲۱.....	فصل دوم پوشه بندی برنامه
۲۲.....	Bin
۲۲.....	Config
۲۳.....	Docs
۲۳.....	Models

۲۳.....	node_modules
۲۳.....	Public
۲۳.....	Routes
۲۳.....	Views
۲۴.....	فایل‌های مهم
۲۴.....	Gitignore
۲۴.....	App.js
۲۴.....	Package.json
۲۵.....	فصل ۳ نگاهی به قسمت‌های مهم
۲۷.....	تنظیم EJS به عنوان template engine
۲۷.....	معرفی فایل‌های مسیر دهی در فایل App.js
۲۸.....	ACCI Support
۲۹.....	نمونه شما (پایگاه داده)
۳۰.....	رمزگذاری (Hashing)، رمز کاربر
۳۲.....	بارگذاری (آپلود) عکس از سمت کلاینت
۳۳.....	ارائه تمامی مقالات نوشته شده یک نویسنده
۳۴.....	پسندیدن یک خاطره (پست)
۳۵.....	انتشار خبر خاطره جدید (پست) در کانال تلگرام سفرنامه بنا به درخواست کاربر
۳۷.....	فصل چهارم پایگاه داده
۳۹.....	کاربر
۴۰.....	خاطره (پست)
۴۲.....	پسندیدن
۴۳.....	دیدگاه
۴۵.....	فصل چهارم انواع کاربران، اعتبار سنجی

فصل ۵ تشریح سایت سفرنامه	۴۸
صفحه ورود کاربر	۴۹
ثبت نام	۵۰
ویرایش اطلاعات کاربری	۵۳
افزودن یک خاطره	۵۵
پسندیدن یک خاطره	۵۶
منابع	۵۷

## مقدمه

در ابتدا لازم است توضیح خلاصه‌ای در مورد سفر بدهیم، اینکه سفر چیست؟ و چه تعریفی برای آن داریم؟ به طور کلی سفر را میتوان مجموعه‌ای از رخدادها بنامیم که در نتیجه‌ی آن یک شخص به تنهایی یا به همراه ۱ یا چند نفر به مدت چند روز در نقطه‌ای غیر از مکانی که در آن ساکن است اقامت می‌کند و در کنار آن نقاط دیدنی آن منطقه را نیز مورد کاوش قرار می‌دهد، بنابراین می‌توان تمامی تصمیمات آدمی را مبنی بر بازدید یک مکان دیدنی و یا حتی یک پارک را به عنوان یک سفر نامید پس لازم است دسته بندی کلی تری برای سفر داشته باشیم؛

### سفرهای درون شهری:

به مجموعه سفرهایی گفته می‌شود که انسان از شهری که در آن اقامت دارد خارج نمی‌شود به عنوان مثال می‌توان گفت یک شهروند تهرانی، کاخ سعدآباد را به عنوان مقصد بازدید خود در یک آخر هفته انتخاب می‌کند. از جمله کسب و کارهایی که در این حوزه میتوان نام برد "پیاده" (البته در حال حاضر این گروه خود را رسانه جوامع و کسب و کارهای شهری نامیده است)، و یا کجارو (زیر مجموعه گروه زومیت) می‌باشند.

### سفرهای برون شهری:

سفرهایی که بنا به تصمیم افراد از ۱ روز تا چندین هفته می‌توانند به طول بی‌انجامند و با توجه به نوع سفر برون شهری، یکسری پیش نیازها نیز باید برای آن فراهم شود. از جمله مهم ترین ارکان سفرهای برون شهری را می‌توان بلیت و رزرو هتل یاد کرد؛ که همانطور که پیش‌تر به آن اشاره کردیم به دلیل سودآوری کلان این دو حوزه تقریباً "سفر" به فراموشی سپرده شده است، بلکه درست نوشته ایم، "سفر"

سفر مجموعه‌ای غنی از تجربه‌های منحصر به فرد است که حتی تجربه گروهی نیز نمی‌تواند تجربه یکسانی برای افراد یک گروه را رقم بزند، شاید این ادعا که تجربه افراد در سفر گروهی شبیه به هم باشد، به توان تایید کرد اما اینکه به صورت کامل یکسان هستند و به عبارت دیگر قابل تطبیق بر روی هم، درست نیست و دقیقاً در همین نقطه بزرگ‌ترین ضعف کسب و کارهای فعال این حوزه خودنمایی می‌کند؛ در حال حاضر هیچ بانک اطلاعاتی از این تجربه‌های منحصر به فرد وجود ندارد، به عبارت دیگر هیچ دفترچه خاطرات سفری یا به عبارت دیگر سفرنامه‌ای وجود ندارد تا کسانی که تجربه ارزشمند سفر را داشته‌اند بتوانند آن را به اشتراک بگذارند تا در کنار این موضوع که سفر به یک منطقه را با تمام سختی و آسانی، زیبایی و تلخی‌اش از نگاه خودشان شرح دهند، بتوانند به کسانی که قصد سفر دارند در انتخاب‌های درست (اعم از انتخاب هتل، وسیله حمل و نقل، مسیر و ...) کمک کنند.

البته بدور از انصاف است، در میان انبوهی از کسب و کارهای فعال این حوزه، لست سکند تنها کسب و کاری بوده است که به این موضوع اهمیت داده و یک قسمت مجزا در سایت خود برای سفرنامه نویسی در نظر گرفته است. اما نکته‌ای که نمیتوان به راحتی از آن گذشت این است که لست سکند یک کسب و کار است و این احتمال وجود دارد که نتواند به عنوان یک رسانه مستقل عمل کند و شاید در مواردی که مربوط به خرید بلیت باشد، این کسب و کار یا اقدام به تحریف و یا از انتشار آن جلوگیری کند.

در نهایت به این نتیجه رسیدیم با ایجاد یک رسانه‌ی مستقل به نام "سفرنامه" تلاش کنیم علاوه بر حفظ عصاره سفرنامه‌ها، از استقلال آن‌ها نیز محافظت کنیم.



## فصل ۱

### ابزارهای مورد استفاده

## محیط برنامه نویسی

ویرایشگر Visual Studio Code یا Vs Code، نرم افزاری است رایگان که توسط شرکت مایکروسافت عرضه شده است و دارای محیطی ساده و ویرایشگری قدرتمند و پیشرفته است و برای کد نویسی و پشتیبانی از C#، C++، CSS، JavaScript، JAVA، Perl، Python، PHP، XML HTML و غیره طراحی و آماده سازی شده است. حجم کم، نصب آسان و برخورداری از تعداد بسیار زیادی از افزونه‌ها برای کامپایل، دیباگ، زیبا سازی نرم افزار و نمایش کدها آن را یک رقیب جدی برای دیگر ویرایشگرها قرار داده است. از این نرم افزار برای ایجاد و عیب‌یابی برنامه‌های رایانش ابری و برنامه‌های وب پیشرفته استفاده میشود. نرم افزار Vs Code برای نسخه‌های مختلف ویندوز، لینوکس و مک عرضه شده است و باعث شده که کاربران بسیاری بدون در نظر گرفتن نوع سیستم عامل بتوانند از این نرم افزار استفاده کنند. در ادامه به تعدادی از ویژگی‌های این نرم افزار اشاره می‌کنیم.

- دسترسی آسان و رایگان
- حجم کم و نصب آسان
- محیط کد نویسی ساده و زیبا
- سازگاری و پشتیبانی از انواع زبان‌های برنامه نویسی
- قابلیت ویرایش و اشکال زدایی دو فایل به صورت همزمان در کنار هم برای مقایسه
- دارا بودن افزونه‌های بسیار برای زبان‌های مختلف
- دارا بودن نقشه کد
- تکمیل خودکار کدها
- جستجوگر هوشمند
- ترمینال دستورات متنی

- نمایش پارامترهای توابع
- امکان پرش مستقیم به توابع و متدها
- نمایش فایل‌های Markdown به فرم کامپایل شده
- امکان استفاده از control Source
- عیب‌یابی سریع و آسان کدها
- قابلیت تغییر ظاهر و قالب نرم افزار به سلیقه کاربر
- دارا بودن صدها توسعه دهنده فعال برای بهبود قابلیت‌های نرم افزار و بروز رسانی سریع و مستمر آن

## زبان برنامه نویسی

جاوا اسکریپت زبان برنامه نویسی پویا و مبتنی بر شی (Object-based) است که هم سمت کاربر و هم سمت سرور، برای ایجاد تعامل با صفحات وب به کار می‌رود. جایی که HTML و CSS ساختار اولیه و ظاهر صفحات وب را تعیین می‌کنند، جاوا اسکریپت نحوه عملکرد صفحات وب را کنترل می‌کند. جاوا اسکریپت، تجربه کاربر را از صفحه وب با تبدیل آن از یک صفحه استاتیک به یک صفحه تعاملی بهبود می‌بخشد. بطور خلاصه، جاوا اسکریپت به صفحات وب رفتار می‌بخشد.

## کجا استفاده می‌شود؟

جاوا اسکریپت عمدتاً برای برنامه‌های مبتنی بر وب و مرورگرهای وب مورد استفاده قرار می‌گیرد. اما جاوا اسکریپت نیز فراتر از وب در نرم افزار، سرورها و کنترل‌های سخت افزاری استفاده می‌شود. در ادامه برخی از مواردی که جاوا اسکریپت در آنها پرکاربرد هستند را نام می‌بریم:

۱. اضافه کردن رفتار تعاملی به صفحات وب

جاوا اسکریپت به کاربران اجازه می‌دهد تا با صفحات وب ارتباط برقرار کنند. تقریباً هیچ محدودیتی برای چیزهایی که می‌توانید با جاوا اسکریپت در یک صفحه وب انجام دهید وجود ندارد. در ادامه به تعدادی از این قابلیت‌ها می‌پردازیم.

- نمایش یا پنهان کردن اطلاعات بیشتر با کلیک یک دکمه
- تغییر رنگ یک دکمه زمانی که ماوس روی آن قرار می‌گیرد

- نمایش اسلاید تصاویر در صفحه اصلی وب
- زوم کردن بر روی یک عکس
- نمایش تایمر یا شمارش معکوس در وب سایت
- پخش فایل صوتی و تصویری در یک صفحه وب
- نمایش انیمیشن‌ها

## ۲. ایجاد برنامه‌های وب و موبایل

توسعه دهندگان می‌توانند از فریم‌ورک‌های مختلف جاوا اسکریپت برای توسعه و ساخت برنامه‌های وب و موبایل استفاده کنند. فریم‌ورک‌های جاوا اسکریپت مجموعه‌ای از کتابخانه‌های جاوا اسکریپت هستند که قابلیت نوشتن کد از قبل را به توسعه دهندگان برای استفاده از ویژگی‌های برنامه‌نویسی روزمره ارائه می‌دهند؛ به معنای واقعی کلمه یک فریم‌ورک برای ساخت وب سایت‌ها یا برنامه‌های کاربردی وب. فریم‌ورک‌های محبوب جاوا اسکریپت React، Angular و Vue می‌باشند. بسیاری از شرکت‌ها از Node.js که یک محیط runtime جاوا اسکریپت بر روی موتور گوگل V8 کروم می‌باشد، استفاده می‌کنند. چند نمونه از مثال‌های مشهور آن: PayPal، LinkedIn، Netflix و Uber می‌باشد.

## ۳. ساخت وب سرورها و توسعه برنامه‌های کاربردی سرور

فراتر از وبسایت‌ها و برنامه‌های کاربردی، توسعه دهندگان همچنین می‌توانند از جاوا اسکریپت برای ساخت وب سرورهای ساده و توسعه زیرساخت‌های back-end توسط Node.js استفاده کنند.

## ۴. توسعه بازی

شما همچنین می‌توانید از جاوا اسکریپت برای ایجاد بازی‌های مرورگر استفاده کنید. این‌ها یک راه عالی برای تمرین مهارت‌های برنامه نویسان مبتدی می‌باشد.

## چرا از جاوا اسکریپت؟

به غیر از امکانات نامحدود، دلایل زیادی برای توسعه دهندگان وب برای استفاده از جاوا اسکریپت در مقایسه با سایر زبان‌های برنامه نویسی وجود دارد:

- جاوا اسکریپت تنها زبان برنامه نویسی بومی برای مرورگر وب است
- جاوا اسکریپت محبوب‌ترین زبان است
- پایین بودن threshold برای شروع

- یک زبان سرگرم کننده برای یادگیری

## بخش سرور

برای بخش سرور از نود جی اس (Node Js) که فریم ورک (Framework) جاوا اسکریپت به حساب می آید، استفاده شده است. در ادامه توضیحاتی در مورد قسمت سرور ارائه می کنیم.

### فریم ورک (Framework)

فریم ورک در لغت به معنی "چارچوب" است و منظور از فریم ورک نرم افزاری، قابلیت است که معمولاً با کمک گرفتن از مجموعه ای از دستورالعمل ها، کتابخانه ها، رابط کاربری، کلاس ها و در کل امکانات فراهم شده از جانب یک نرم افزار یا یک زبان برنامه نویسی، ساختاری جدید ایجاد می کند تا به کارگیری آن نرم افزار یا زبان برنامه نویسی در راستای اهداف ایجاد فریم ورک، ساده تر و نتیجه با سهولت بیشتری محقق گردد، به عبارتی هدف از ایجاد فریم ورک ها معمولاً فراهم نمودن ساز و کاری است که بتوان از امکانات یک نرم افزار یا زبان برنامه نویسی به نحو سریع تر و در برخی موارد کامل تر و حرفه ای تر استفاده کرد، فریم ورک ها در مقایسه با کتابخانه های نرم افزاری، به لحاظ مفهومی از وسعت بیشتری برخوردارند.

### نود جی اس (Node Js)

Node.js یک پلتفرم سمت سرور مبتنی بر موتور جاوا اسکریپت گوگل کروم (V8 Engine) می باشد. Node.js

تمام چیزهایی که برای اجرای یک برنامه نوشته شده به زبان جاوا اسکریپت را نیاز دارید برایتان فراهم می کند. آقای Ryan Dahl در سال ۲۰۰۹ Node.js را معرفی کرد تا نشان دهد جاوا اسکریپت قدرتمندتر از این حرف ها است که فقط برای پویا سازی صفحات وب در فرانت اند استفاده شود. در واقع به کمک Node.js زبان برنامه نویسی جاوا اسکریپت به جای اجرا در مرورگر در محیط سرور اجرا می شود Node.js به شما اجازه می دهد به آسانی و سادگی برنامه های تحت شبکه مقیاس پذیر و بزرگ بنویسید.

چرا نود جی اس

- زمانی که فعالیت سرور به شدت افزایش پیدا می کند مانند زمان DDOS Attack این فریم ورک تمامی این سنگینی را به دوش کشیده و هیچ رد پای برای متجاوزین باقی نمی گذارد. بدین ترتیب اولین ویژگی آن امنیت و نفوذ ناپذیری به آن است.
- همانطور که گفتیم نود جی اس به دلیل تک رشته ای و نامتقارن بودن، در زمان پردازش داده های ورودی و خروجی (I/O) دیگر فرآیندها را متوقف نمی کند به همین دلیل پاسخگویی آن در یک چشم بر هم زدن انجام می شود. در واقع شما به طور همزمان می توانید چندین Task مثل پیام دادن، کپی کردن فایل ها از روی دیسک، کار بر روی دیتابیس و ... را بر روی سرور خود اجرا کنید.
- همانطور که گفتیم در اکثر پلتفرم ها برای هر درخواست یک رشته (Thread) ایجاد می شود که در واقع هر کدام از آنها فضایی را بر روی RAM سرور (مثلا ۵۰۰ مگابایت) اشغال می کنند که طبیعتا در صورت ایجاد درخواست های بیش از حد RAM به حالت Shout down درخواهد آمد. در حقیقت کدهای مربوط به جاوا اسکریپت در رشته اصلی (main Thread) و کدهای مربوط به قسمت I/O در رشته های دیگر پردازش می شوند.
- موتور مجازی گوگل V8 در جاوا اسکریپت که وظیفه اجرا کدهای جاوا را برعهده دارد در پس از دریافت کدها و در زمان اجرا، عملیات شبیه سازی آن ها را انجام می دهد بدین معنی که زبان کدها را تبدیل به زبان کدهای موتور مجازی V8 می کند. بدین ترتیب توابعی که زیاد درخواست می شوند به علت

شباهت به کدهای ماشین سریع تر پردازش شده و در نتیجه عملکرد کلی کدها سریع تر انجام می گیرد.

- همانطور که توضیح دادیم این پلتفرم به صورت متن باز در اختیار عموم قرار دارد و برنامه نویسان بسیاری در توسعه آن نقش داشته اند بدین معنی که یک کتابخانه بسیار وسیع و مجهز برای آن وجود دارد که به شما این امکان را می دهد تا برای اکثر مفاهیم ابتدایی و پیش پا افتاده از آن استفاده کنید و دیگر مجبور به کد نویسی درباره کوچکترین چیزها نباشید.

در ادامه به برخی از کتابخانه و ماژول های (افزونه) استفاده شده در نود جی اس می پردازیم.

## Express.js

اکسپرس جی اس (Express js) ، یا بطور ساده، اکسپرس (Express) یک فریم ورک انعطاف پذیر وب برای نود جی اس است که مجموعه ای از ویژگی ها همچون API های ساده در اختیار قرار می دهد تا در ساختن بخش مدیریت (بک-اند یا back-end) در یک وبسایت یا اپلیکیشن موبایل از آنها استفاده کنیم. اکسپرس متن باز است و در حال حاضر توسط موسسه نود جی اس پشتیبانی می شود. اکسپرس (Express js) همچنین میان افزارها (middlewares) و تعاریف مسیرنویسی (Routing) پایه را برای ما فراهم می سازد.

## Passport.js

Passport یک میان افزار برای نود جی اس است. از آنجایی که این ابزار بسیار منعطف و ماژولار است می تواند در اپلیکیشن های مبتنی بر وب Express قرار بگیرد Passport می تواند احراز هویت را از طریق نام کاربری و رمز عبور، فیسبوک، توییتر و... برای برنامه ما انجام دهد.

## jalali-moment

این پلاگین (افزونه) برای سهولت هرچه بیشتر در کار با تاریخ شمسی در جاوا اسکریپت یا تایپ اسکریپت (ts) ایجاد شده است همچنین اگر میخواهید یک تقویم میلادی را به تقویم شمسی تبدیل کنید این کتابخانه مناسب ترین گزینه است.

## Fs

از این ماژول برای کار بر روی فایل استفاده می‌کنیم، اعم از حذف، ایجاد و ذخیره و به عبارت دیگر توابع

:POSIX

- Files and Directories.
- Advanced File Operations.
- Processes.
- Long Jumps.
- Signal Handling.
- Obtaining Information at Runtime.
- Terminal I/O.
- Process Groups and Job Control.

## bcrypt js

استفاده از bcrypt یک راه امن برای ذخیره کلمه عبور در پایگاه داده است، به زبان ساده‌تر، برای امنیت بیشتر پیش از ذخیره رمز عبور کاربران یا اطلاعات حساس در پایگاه داده، آن را با کمک این ماژول (کتابخانه) و با تولید بایت‌های تصادفی (The salt) آن را رمزگذاری کرده و سپس ذخیره می‌کنیم و به تبع آن برای استفاده از رمز (به عنوان مثال زمانی که کاربر برای ورود به صفحه خود رمز خود را وارد می‌کند) از همین ماژول بدون نیاز به رمزگشایی استفاده می‌کنیم.



## Multer

یک میدلور (میان افزار) که برای آپلود (بارگذاری) فایل ها (multipart/form-data) از سمت کلاینت به سرور به کار می رود.

## querystring

از این ماژول برای قابلیت پشتیبانی URL به زبان پارسی در Node JS استفاده می کنیم.

## Mongoose

Mongoose یک چارچوب (ماژول) JavaScript است که معمولاً در یک برنامه Node.js با یک بانک اطلاعاتی MongoDB استفاده می شود و یک Mapper Document Object (DOM) است.

### مونگو دی بی (Mongo DB)

مونگو دی بی (Mongo DB) یکی از معروف ترین پایگاه داده های No SQL است که ساختار منعطفی دارد و بیشتر در پروژه هایی با حجم بالای داده استفاده می شود. این پایگاه داده پلتفرمی متن باز و رایگان است و با مدل داده های مستندگرا (Document - Oriented) کار می کند و در ویندوز، مکینتاش و لینوکس قابل استفاده است؛ برخی ویژگی های مونگو دی بی عبارت است از:

- مونگو دی بی به علت مستندگرا بودن مدل ذخیره داده ها در مقایسه با دیتابیس های رابطه ای بسیار منعطف تر و مقیاس پذیرتر است و بسیاری از نیازمندی های کسب و کارها را برطرف می کند.
- این پایگاه داده برای تقسیم داده ها و مدیریت بهتر سیستم از شاردینگ (Sharding) استفاده می کند. شاردینگ به معنی تکه تکه کردن است و در لود بالای شبکه انجام می شود. به گونه ای که دیتابیس به چند زیربخش تقسیم می شود تا روند پاسخ دهی به درخواست هایی که از سمت سرور می آید، راحت تر شود.

- داده‌ها با دو کلید اولیه و ثانویه قابل دسترسی هستند و هر فیلدی قابلیت کلید شدن را دارد. این امر زمان دسترسی و پردازش داده را بسیار سریع می‌کند.
- همانند سازی (Replication) یکی دیگر از خصوصیات مهم مونگو دی بی است. در این تکنیک از یک داده به عنوان داده اصلی کپی‌هایی تهیه شده و بخش‌های دیگری از سیستم پایگاه داده ذخیره می‌شود. در صورت از بین رفتن و یا مخدوش شدن این داده، داده‌های کپی شده به عنوان داده اصلی و جایگزین مورد استفاده قرار می‌گیرند.

دیتابیس‌های رابطه‌ای دارای اسکیم (Schema) هستند. یعنی ساختار خاصی برای داده‌ها در نظر گرفته و مدل‌های محدودی را ذخیره می‌کنند. اما مونگو دی بی و به طور کلی دیتابیس‌های NoSQL در برابر پذیرش داده‌هایی با تنوع مختلف بسیار منعطف هستند و این مزیت مهمی برای برنامه نویسان محسوب می‌شود. مقیاس پذیری این پایگاه داده باعث استفاده از آن در پروژه‌هایی می‌شود که با کلان داده‌ها (Big Data) سروکار دارند. علاوه بر مزایای گفته شده مشکلاتی نیز در مونگو دی بی وجود دارد که ممکن است دردسر ساز شود. این پایگاه داده در استفاده از کلید خارجی (Foreign Key) برای داده‌ها ضعف دارد و ممکن است پایداری داده‌ها و یکپارچگی سیستم را به هم بریزد. همچنین در خوشه بندی داده‌های موجود در این پایگاه داده، تنها می‌توان یک گره (Node) را به عنوان گره اصلی (Master) انتخاب کرد که اگر از بین برود، ممکن است مرتب سازی زیرگره‌های آن از بین برود. این مشکل در پایگاه داده کاساندر (Cassandra) برطرف شده است.

## کلاینت

### HTML

اگر بخواهیم یک ترجمه‌ی فارسی از HTML داشته باشیم، می‌توانیم به آن، زبان نشانه گذاری فرا متن بگوییم. HTML ابزاری است که با استفاده از چیزهایی به نام تگ‌ها، اقدام به ساخت قالب کلی صفحات وب میکند. به این معنا که شما با استفاده از HTML می‌توانید فوندانسیون و اسلکت اصلی وبسایت خود را طراحی کنید.

## CSS

به صورت خلاصه، CSS تعیین می‌کند هر کدام از عناصر HTML چگونه نمایش داده شوند. در واقع به کمک آن می‌توان برای هر عنصر از صفحه، سبک نمایشی خاصی ایجاد کرد. با این ابزار به اجزای صفحه‌ی وب، رنگ و لعاب می‌دهیم تا صفحات زیباتری داشته باشیم!

## JQuery

JQuery یک کتابخانه‌ی سریع و کوچک از نوع جاوا اسکریپت است که قابلیت‌های زیادی برای کاربر فراهم می‌کند. این کتابخانه کارهایی مثل ایجاد و ویرایش متون HTML، کنترل رویدادها، استفاده از انیمیشن و آژاکس را بسیار برای ما راحت می‌کند. همچنین یک رابط کاربری دارد که کار با آن ساده است و در بسیاری از مرورگرها مورد استفاده قرار می‌گیرد. جی کوئری با انعطاف پذیری و گسترش پذیری منحصر به فردی که دارد، شیوه‌ی نوشتن جاوا اسکریپت را برای میلیون‌ها کاربر در فضای وب تغییر داده است.

## Bootstrap

Bootstrap مجموعه‌ای از ابزارهای رایگان برای ایجاد صفحات وب و نرم افزارهای تحت وب است که شامل دستورات HTML، CSS و توابع جاوا اسکریپت جهت تولید و نمایش فرم‌ها، دکمه‌ها، تب‌ها، ستون‌ها و سایر المان‌های مورد نیاز طراحی وب می‌باشد. Bootstrap در ابتدا توسط مارک اتو و جاکوب تورنتون و در جهت ایجاد یک چارچوب ظاهری مشخص و یکسان در ابزارهای توییتر طراحی و نوشته شد. قبل از شروع این پروژه نمونه‌های زیادی با همین رویکرد ایجاد شده بود که همگی با سرنوشتی مشابه و عدم استقبال طراحان وب دنیا مواجه شده بودند. به دلیل وجود مشکلات اساسی در نمونه‌های دیگر، سازنده اصلی توییتر یا همان مارک اتو تصمیم به ساخت یک سیستم داخلی و قدرتمند برای خود با نام Bootstrap گرفت.

## EJS

EJS یک زبان ساده الگویی است که به شما امکان تولید نشانه گذاری HTML با JavaScript ساده را می‌دهد.

## MVC

Model View Controller یک معماری برنامه نویسی است که نرم افزار را به سه بخش مجزا تقسیم می کند. این بخش ها عبارت اند از Model برای ارتباط با پایگاه داده، View برای ایجاد یک رابط کاربری برای نمایش و تغییر در داده های پایگاه داده و Controller برای پردازش درخواست های کاربر می باشد.

### ۱. Model

در mvc مدل ها همان کلاس های برنامه هستند که شبیه به دیتابیس هستند ولی می توانند متفاوت و بیشتر باشند. همچنین Model وظیفه تایید داده ها را برعهده دارد تا اطلاعات درست درون آنها قرار بگیرد و برای دیتابیس ارسال شود.

### ۲. View

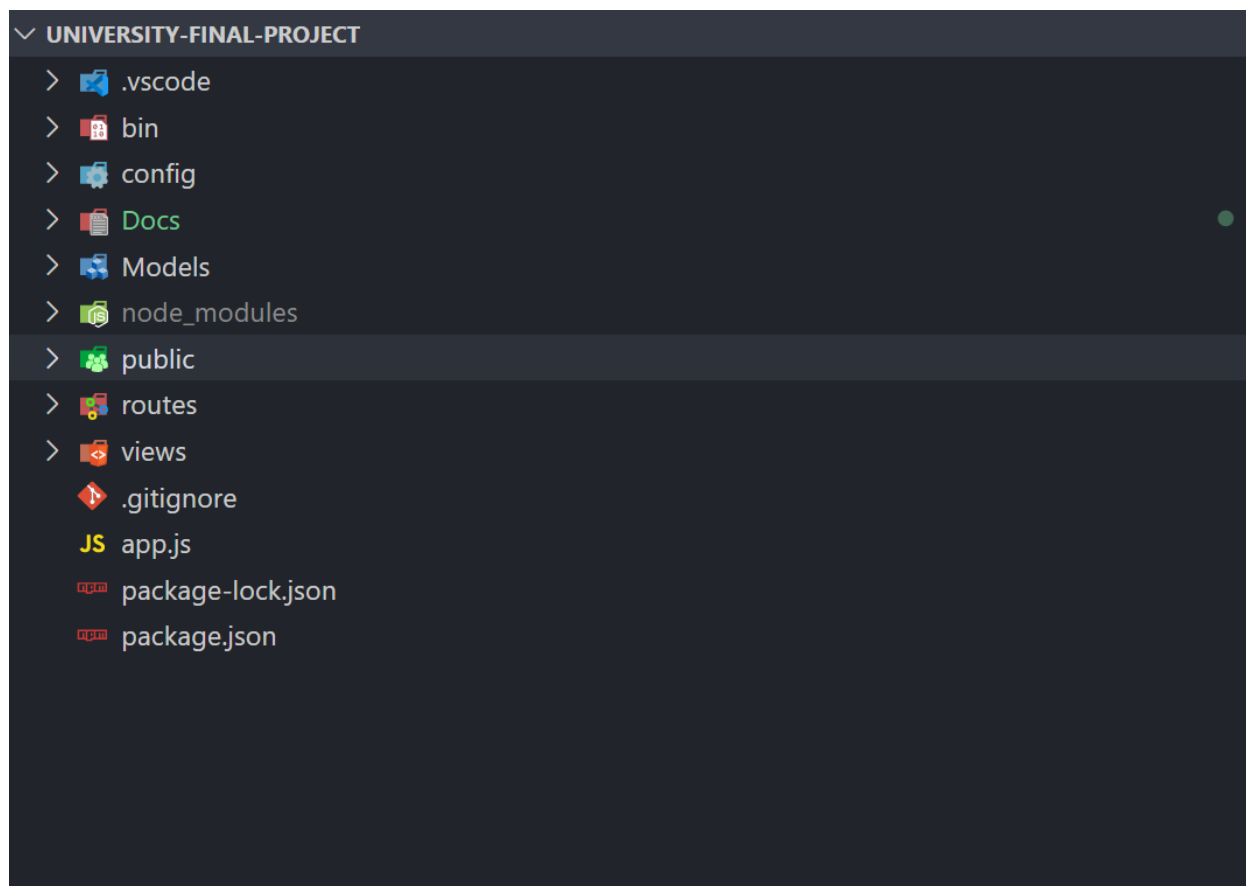
این بخش اطلاعات را به کاربر نشان می دهد، ویو کاری به منطق برنامه ندارد و فقط یک model دریافت می کند و آن را به هر شکلی که نیاز باشد به کاربر نشان می دهد، این model هم توسط کنترلر ایجاد شده است و برای view ارسال شده است. نکته مهمی که در بخش View باید مد نظر داشت این است که این بخش مسئول کنترل صحت داده های وارد شده از طریق کاربر و همچنین مسئول صحت داده های نشان داده شده به کاربر نیست. در واقع این بخش با داده های خام کار میکند و مسولیت صحت داده با model است.

۳. اطلاعات را از دیتابیس در قالب model هایی می گیرد و بعد پردازش خود را روی آن انجام می دهد و اطلاعات را در قالب model جدیدی به view می فرستد تا ویو هر طور خواست این اطلاعات را به کاربر نشان دهد. فرض کنید که ما برای صفحه پروفایل یک view برای موبایل داریم و یک view برای ویندوز، حال کنترلر بنا به تشخیص که کاربر با موبایل است یا نه یک view را انتخاب می کند تا برای کاربر ارسال شود و می بینید که یک اطلاعات مشخص در دو view متفاوت ظاهر می شود و view (ظاهر) از منطق برنامه جدا است.

## فصل ۲

### پوشه بندی برنامه

ساختار کلی پوشه بندی پروژه سفرنامه به شکل زیر است.



تصویر ۱

### Bin

در پوشه فایل WWW قرار گرفته که اطلاعات اولیه همچون کدهای لازم برای راه اندازی پروژه (همچون تعریف پورت ۳۰۰۰ بنا به نیاز برنامه نویس) در آن قرار گرفته است.

### Config

در این پوشه فایل‌های پیکربندی پروژه را قرار داده‌ایم؛ فایل‌هایی همچون DB.js که آدرس مورد نیاز برنامه برای اتصال به پایگاه داده را در آن تعریف کرده‌ایم.

## Docs

در این پوشه فایل‌های مستند مربوط به پروژه نگهداری می‌شود.

## Models

در این پوشه فایل‌های مربوط به شما (مدل) پایگاه داده نگهداری می‌شود.

## node\_modules

به صورت کلی این پوشه تمامی ماژول‌هایی را که به صورت جداگانه بنا به پیش فرض یا درخواست برنامه نویس بر روی پروژه نصب شده‌اند، قرار دارد از این پوشه میتوان اصطلاحاً کش برای ماژول‌های خارجی نیز نام برد. عمده حجم پروژه را نیز (به استثنا بخش چند رسانه‌ای) این قسمت تشکیل می‌دهد.

## Public

این پوشه محل نگهداری فایل‌های چند رسانه‌ای و در کنار آن فایل‌های CSS و Java Script برای کلاینت، می‌باشد.

## Routes

در برنامه‌هایی که با فریم ورک‌هایی همچون نود جی اس و فریم ورک اکسپرس توسعه داده شده، درخواست‌های فرستاده شده از url را قسمت‌های خاصی از برنامه بررسی می‌کنند. اینکه کدام درخواست در کدام قسمت از برنامه باید بررسی بشود را روترها مشخص می‌کنند. در واقع مسیر یابی درخواست‌ها را روترها انجام می‌دهند. اگر درخواستی داشته باشیم که مسیری برای آن وجود نداشته باشد با خطای ۴۰۴ مواجه می‌شویم. تصویر زیر نمونه خطای ۴۰۴ را که از سایت آکادمی آی تی گرفته ایم را نشان می‌دهد. در این پوشه بنا به نیاز پروژه دو فایل آدرس دهی قرار داده شده است.

## Views

در این پوشه، فایل‌های کلاینت، به طور دقیق‌تر فایل‌های HTML قرار دارد.

## فایل‌های مهم

### **Gitignore**

هدف از این فایل برای آپلود و یا عدم آپلود فایل‌های پروژه در گیت لب یا گیت هاب می‌باشد به عنوان مثال ما در این فایل تعیین کردیم فایل‌های چند رسانه‌ای در گیت هاب آپلود نشوند.

### **App.js**

اگر فایل `www` اطلاعات اولیه را برای راه اندازی پروژه فراهم می‌کند، فایل `App` برای راه اندازی پروژه به کار می‌رود و لازم مواردی از قبیل راه اندازی پایگاه داده یا اتصال برنامه به پایگاه داده نیز در این فایل قرار بگیرد.

### **Package.json**

در این فایل نام تمامی ماژول‌های نصب شده و ورژن آن‌ها قرار گرفته است.



## فصل ۳

نگاهی به قسمت‌های مهم

```
DB.js:
module.exports = 'mongodb://127.0.0.1:27017/SafarNameDatabase';

App.js:
const mongoose = require('mongoose');
const DbUrl = require("./config/DB");
mongoose.set('useCreateIndex', true);
mongoose.connect(DbUrl, {
  useNewUrlParser: true,
  useUnifiedTopology: true
})
.then(() => console.log('MongoDB Connected.'))
.catch(err => {
  console.log(err);
  next(err);
});
```

## تصویر ۲

در فایل DB.js آدرسی که برنامه باید به پایگاه متصل شود قرار گرفته (علاوه بر آدرس، نام پایگاه داده مورد نظر نیز تعیین گردیده است).

در فایل App.js با کمک از mongoose به پایگاه داده متصل شدیم و در ادامه برای اتصال موفق یک پیام نیز در نظر گرفته شده است.

تنظیم EJS به عنوان template engine



```
app.set('views', path.join(__dirname, 'views'));  
app.set('view engine', 'ejs');
```

تصویر ۳

معرفی فایل‌های مسیر دهی در فایل App.js



```
var indexRouter = require('./routes/index');  
var usersRouter = require('./routes/users');  
  
app.use('/', indexRouter);  
app.use('/', ensureAuthenticated, usersRouter);
```

تصویر ۴



```
let unescape = require('querystring').unescape;
app.use((req, res, next) => {
  req.url = unescape(req.url);
  next();
});
```

تصویر ۵

متأسفانه در حال حاضر اکسپرس از ریکوئست به طور دقیق‌تر آدرس به با کارکتر فارسی پشتیبانی نمی‌کند ( به عنوان مثال، “علی همایونی/articlesBy/localhost:۳۰۰۰”) برای حل این مشکل از کد بالا استفاده می‌کنیم. همچنین در قسمت بررسی url ها نیز از کد زیر کمک می‌گیریم.



```
router.get("/articlesBy/:word(("[\\u0600-\\u06FF]+\\s?)+$)", (req, res) => {
  .
  .
  .
})
```

تصویر ۶

```
const mongoose = require('mongoose');
const UserSchema = new mongoose.Schema({
  userName: {
    type: String,
    required: true
  },
  email: {
    type: String,
    required: true
  },
  password: {
    type: String,
    required: true
  },
  profileImage: {
    type: String,
    required: true,
    default: '/images/profileImages/defaultImage/profile.png'
  },
  description: {
    type: String,
    default: "یه ملجرا جو"
  },
  lastActivity: {
    type: Date,
    default: Date.now
  },
}, {
  timestamps: true
});
module.exports = mongoose.model('User', UserSchema);
```

در عکس بالا یکی از شمای‌های (مدل‌های) استفاده شده پروژه را مشاهده می‌کنید، این مدل که برای کاربر برنامه طراحی شده شامل مواردی چون نام، آدرس رایانامه، رمز عبور، تعریف از خود و آخرین فعالیت می‌باشد؛ که از این بین در زمان ایجاد یک کاربر تکمیل مواردی که ویژگی `required: true` را شامل هستند، الزامی است. همچنین این مدل قابلیت بروز رسانی خودکار در زمان را نیز دارد یعنی در صورتی که اطلاعات کاربر تغییر کند، به عنوان مثال کاربر رمز خود را تغییر دهد، تاریخ این تغییر در آبجکت پایگاه داده‌اش ذخیره می‌گردد، این قابلیت توسط timestamps اجرا می‌شود.

### رمزگذاری (Hashing)، رمز کاربر

```
async function hashPassword(newPasswordSendForHashing) {
  const hashedPassword = await new Promise((resolve, reject) => {
    bcrypt.hash(newPasswordSendForHashing, 10, function (err, hash) {
      if (err) reject(err)
      resolve(hash)
    });
  })
  return hashedPassword
}
```

#### تصویر ۸

با کمک پکیج `bcrypt` رمز کاربر یا به طور کلی اطلاعات حساس را رمز گذاری می‌کنیم و سپس در پایگاه داده ذخیره می‌کنیم؛ همچنین در زمانی که به این اطلاعات نیاز داشته باشیم به عنوان مثال در زمانی که کاربر قصد ورود به برنامه را داشت و برنامه باید رمز ورودی او را با رمزی که در پایگاه داده ذخیره شده راستی آزمایی کند؛ برای حفظ دو چندان امنیت و همچنین جلوگیری از تحمیل بار اضافی به سیستم، این اطلاعات در زمان نیاز،

رمز گشایی نخواهند شد بلکه در همان حالت با اطلاعات مورد نظر ما مقایسه خواهد شد، نمونه از این حالت را در کد زیر مشاهده می‌کنیم.



```
bcrypt.compare(password, user.password, (err, isMatch) => {  
  if (isMatch) {  
    return done(null, user)  
  } else {  
    return done(null, false, {  
      message: 'رمز اشتباه است'  
    })  
  }  
})
```

تصویر ۹

## بارگذاری (آپلود) عکس از سمت کلاینت

```
const multer = require("multer");
let uploadProfileImage = multer({
  storage: multer.diskStorage({
    destination: function (req, file, callback) {
      callback(null, "public/images/profileImages");
    },
    filename: function (req, file, callback) {
      callback(
        null,
        file.fieldname + "-" + Date.now() + path.extname(file.originalname)
      );
    }
  }),
  fileFilter: function (req, file, callback) {
    if (
      path.extname(file.originalname) !== ".png" &&
      path.extname(file.originalname) !== ".gif" &&
      path.extname(file.originalname) !== ".jpg" &&
      path.extname(file.originalname) !== ".jpeg"
    ) {
      return callback(null, false);
    }
    callback(null, true);
  },
  limits: {
    fileSize: 7000000
  }
});
```

### تصویر ۱۰

کد بالا برای بارگذاری عکس پروفایل کاربر با کمک افزونه multer نوشته شده است، در قسمتی از آن مقصد نهایی ذخیره فایل در هاست تعریف شده، در قسمتی دیگر محدودیت‌های نوع فایل نوشته شده که هدف از آن قبول تنها فایل‌های از نوع عکس است؛ در قسمت دیگر حداکثر حجم قابل قبول تعریف شده است.



ارائه تمامی مقالات نوشته شده یک نویسنده

```
router.get("/authorArticles/:id", (req, res) => {
  let postKey = [];
  let variable = 0;
  for (let i = 0; req.path[i] != null; i++) {
    if (req.path[i] === "/" ) {
      variable++;
    }
    if (variable === 2) {
      if (req.path[i] !== "/" ) {
        postKey.push(req.path[i]);
      }
    }
  }
  postKey = postKey.join("");
  User.findOne({
    userName: postKey
  })
  .then(user => {
    Post.find({
      author: user._id
    })
    .then(posts => {
      lastActivity = moment(user.updatedAt, 'YYYY/MM/DD').locale('fa').format('YYYY/MM/DD');
      posts.forEach(post => {
        post["time"] = moment(post.createdAt, 'YYYY/MM/DD').locale('fa').format('YYYY/MM/DD');
      });
      res.render('authorArticles', {
        user,
        posts,
      });
    });
  })
});
```

تصویر ۱۱

در این قسمت، ابتدا آی دی کاربر از url که دریافت کردیم استخراج می‌شود و در مرحله بعد در پایگاه داده جستجو کرده و در نهایت تاریخ‌های میلادی را به شمسی تبدیل و به کلاینت ارسال می‌کنیم.

## پسندیدن یک خاطره (پست)

```
router.post("/addLike/:id", (req, res) => {
  let postKey = [];
  let variable = 0;
  for (let i = 0; req.path[i] != null; i++) {
    if (req.path[i] === "/") {
      variable++;
    }
    if (variable === 2) {
      if (req.path[i] != "/") {
        postKey.push(req.path[i]);
      }
    }
  }
  postKey = postKey.join("");
  Like.findOne({
    post: postKey
  })
  .then(likeInformation => {
    if (!likeInformation) {
      const newLikeInformation = new Like({
        post: req.body.postId,
        users: [req.user._id],
        counted: 1
      });
      req.flash("success_msg", "شما این خاطره را پسندیدی");
      newLikeInformation.save();
    }
    else {
      const checkUserLikedPost = likeInformation.users.find(element => element == req.user.id);
      if (checkUserLikedPost) {
        req.flash("error_msg", "قبلا این خاطره را پسندیدی");
      }
      else {
        req.flash("success_msg", "شما این خاطره را پسندیدی");
        likeInformation.users.push(req.user._id);
        likeInformation.counted = likeInformation.counted + 1;
        likeInformation.save();
      }
    }
  })
  res.redirect(`/article/${req.body.postId}`);
});
```

## تصویر ۱۲

در این قسمت ابتدا url را گرفته و قسمتی که در آن id خاطره قرار گرفته استخراج می‌کنیم؛ در مرحله بعد با کمک id، خاطره (پست) ابجکت لایک مورد نظر را از پایگاه داده استخراج می‌کنیم در صورتی که تا به حال این خاطره توسط کسی پسندیده نشده باشد، یک ابجکت جدید ساخته می‌شود همچنین در صورتی که قبلا ایجاد

شده باشد، برنامه چک می‌کند که آیا کاربر دوباره پسندیده است یا خیر که در صورتی که قبلاً پسندیده باشد، از پسندیدن مجدد آن جلوگیری کند؛ در غیر اینصورت به قسمت شمارنده (counter) آجکت لایک، یک عدد اضافه و id کاربر به آرایه‌ای که کاربرانی که خاطره را پسندیده‌اند اضافه می‌شود. لازم به ذکر است کاربرانی که وارد حساب کاربری خود نشده‌اند اجازه پسندیدن یک خاطره را ندارند.

انتشار خبر خاطره جدید (پست) در کانال تلگرام سفرنامه بنا به درخواست کاربر

```
module.exports = {  
  cookieKey: 'SafarNameCookieKey',  
  telegram_api_token: '1299772442:AAHRCeo0DCzUkKwLP0chKZwhWwPowTlaIHo'  
};
```

تصویر ۱۳

```
const { telegram_api_token } = require('../config/securityKeys');  
const { Telegraf, Markup, Extra } = require('telegraf')  
const bot = new Telegraf(telegram_api_token)
```

تصویر ۱۴

```

if (req.body.shareToTelegramChannel) {
  const keyboard = Markup.inlineKeyboard([
    Markup.urlButton("بزن بریم", `https://www.zoomit.ir/`, 0),
  ]);
  bot.telegram.sendMessage(
    '@SafarNameThePlaceForWriteAMemory',
    `
    -----
    ✂✂✂✂✂\n
    -----
    \n خاطره ای جدید منتشر شد \n
    ${moment(new Date(), 'YYYY/MM/DD').locale('fa').format('YYYY/MM/DD')} \n
    ${newPost.subject}\n
    ${newPost.summery}\n
    به قلم ${req.user.userName} \n
    -----
    \n✂✂✂✂✂\n`,
    Extra.markdown().markup(keyboard)
  );
}

```

## تصویر ۱۵

در قسمت انتشار خاطره جدید یک گزینه اختیاری برای کاربر در نظر گرفته شده است و آن انتشار خبر خاطره جدید در کانال سفرنامه است، برای اینکار در ابتدا باید یک ربات در تلگرام ایجاد و کلید دسترسی به آن را از تلگرام دریافت کنیم (عکس ۱۳)؛ در مرحله بعد بایستی با کمک افزونه (ماژول) telegraf به ربات تلگرام متصل شویم، همچنین باید در تلگرام یک کانال ایجاد کنیم و ربات مورد نظرمان را به عنوان ادمین تنظیم کنیم. و در مرحله آخر یک پیام جدید را به واسطه ربات در کانال منتشر کنیم.

فصل چهارم

پایگاه داده

همانطور که قبل تر عنوان شد، برای این پروژه از پایگاه داده مونگو دی بی استفاده شده است. در سفرنامه، ۴ شما (مدل) به ترتیب کاربر، خاطره (پست)، پسندیدن، دیدگاه استفاده شده است که در ادامه هر کدام را توضیح می دهیم. همچنین برای تمامی پراپرتی (اجزای) شما در صورت لزوم قابلیت در نظر گرفته شده به نام تریم که هدف از استفاده آن این است که در صورتی که کاربر در زمان نوشتن یک قسمت خواسته یا ناخواسته در ابتدا یا انتهای آن فاصله (space) گذاشت trim این فاصله زائد را حذف می کند.

```
const mongoose = require('mongoose');

const PostSchema = new mongoose.Schema({
  subject: {
    type: String,
    required: true,
    trim: true
  },
  {
    timestamps: true
  }
});

module.exports = mongoose.model('Post', PostSchema);
```

تصویر ۱۶

```

const mongoose = require('mongoose');
const UserSchema = new mongoose.Schema({
  userName: {
    type: String,
    required: true
  },
  email: {
    type: String,
    required: true
  },
  password: {
    type: String,
    required: true
  },
  profileImage: {
    type: String,
    required: true,
    default: '/images/profileImages/defaultImage/profile.png'
  },
  description: {
    type: String,
    default: "یه ملجرا جو"
  },
  lastActivity: {
    type: Date,
    default: Date.now
  },
},
{
  timestamps: true
});
module.exports = mongoose.model('User', UserSchema);

```

تصویر ۱۷

اطلاعات که ما برای کاربر در نظر گرفتیم نام کاربری، آدرس رایانامه، رمز عبور، عکس پروفایل و توضیح در مورد خود می‌باشد؛ از موارد یاد شده وجود نام کاربری، آدرس رایانامه و رمز عبور در زمان ایجاد یک کاربر جدید در

پایگاه داده الزامی است و باید دارای مقدار باشد؛ مورد عکس و تعریف از خود، این اختیار به کاربر داده شده است تا عکسی بارگذاری یا توضیحی بدهد، در غیر اینصورت طبق برنامه ریزی انجام گرفته توسط برنامه نویس مقادیر پیش فرض برای دو مورد یاد شده، تعریف شده است.

### خاطره (پست)

```
const mongoose = require('mongoose');

const PostSchema = new mongoose.Schema({
  subject: {
    type: String,
    required: true
  },
  article: {
    type: String,
    required: true
  },
  summery: {
    type: String,
    required: true,
  },
  image: {
    type: String,
    required: true,
    default: '/images/postImages/defaultImage/defaultImage.jpg'
  },
  articleKeys: {
    type: Array,
    default: ["سفر"]
  },
  author: {
    type: mongoose.Schema.Types.ObjectId,
    ref: 'User'
  }
}, {
  timestamps: true
});

module.exports = mongoose.model('Post', PostSchema);
```

تصویر ۱۸



برای شما (مدل) سفرنامه (پست) مواردی چون عنوان، متن، خلاصه متن، عکس، کلید واژه و نویسنده در نظر گرفته شده است که برای ایجاد یک پست در پایگاه داده تکمیل تمامی موارد به غیر از کلید واژه الزامی و عکس است؛ همچنین برای عکس و کلید واژه مقدار پیش فرض تعریف شده است. همچنین برای نویسنده از کلید خارجی استفاده شده بدین صورت که کلید (Id) کاربر در آن ذخیره می‌شود.

```

const mongoose = require('mongoose');

const LikeSchema = new mongoose.Schema({
  counted: {
    type: Number,
    required: true,
    default: 0
  },
  users: {
    type: Array,
    required: true
  },
  post: {
    type: mongoose.Schema.Types.ObjectId,
    ref: 'Post'
  }
}, {
  timestamps: true
});
module.exports = mongoose.model('Like', LikeSchema);

```

تصویر ۱۹

برای شما (مدل) پسندیدن از مواردی چون شمارنده، برای شمردن تعداد کاربرانی که یک پست خاص را پسندیده‌اند، کاربرانی که پست را پسندیده‌اند و در آخر پست مدنظر ما (id آن) که در قالب یک کلید خارجی در

این شما ذخیره می‌شود؛ شمارنده بصورت پیش فرض مقدار صفر در خود ذخیره کرده است و هربار که کاربری جدید روی دکمه پسندیدن فشار (کلیک) می‌کند یک واحد به آن اضافه می‌شود.

## دیدگاه

```
const mongoose = require('mongoose');

const CommentSchema = new mongoose.Schema({
  comment: {
    type: String,
    required: true,
  },
  userName: {
    type: String,
    required: true,
  },
  post: {
    type: mongoose.Schema.Types.ObjectId,
    ref: 'Post'
  }
}, {
  timestamps: true
});
module.exports = mongoose.model('Comment', CommentSchema);
```

تصویر ۲۰

این شما شامل ۳ قسمت شرح دیدگاه، نام کاربر و پست که به عنوان یک کلید خارجی تعریف شده، می باشد که در زمان ایجاد یک دیدگاه، دو قسمت آن یعنی کلید خارجی پست و کاربر توسط برنامه و متن دیدگاه توسط کاربر تکمیل می شود.

## فصل چہارم

انواع کاربران، اعتبار سنجی

به طور کلی سه گروه کاربر وجود دارد، کاربران آزاد (یا مهمان)، کاربرانی که وارد حساب کاربری شده‌اند و در انتها کاربر ادمین. با توجه به این گروه بندی بایستی یکسری محدودیت دسترسی برای کاربران در نظر بگیریم به عنوان مثال کاربری که وارد حساب کاربری‌اش نشده است، نباید به قسمت افزودن خاطره دسترسی داشته باشد و یا کاربران (مهمان) امکان پسندیدن یک پست (خاطره) را ندارند و همچنین کاربران مهمان و کاربرانی که وارد حساب کاربری شده‌اند نباید به امکانات ادمین دسترسی پیدا کنند. برای ایجاد و کنترل پیوسته این محدودیت‌ها و همچنین جلوگیری از شلختگی، کدهای مسیره‌دهی (Routing) را به سه قسمت تقسیم شده است، قسمت اول (فایل indexRouter)، مسیرهایی است که تمامی کاربران بدون توجه به اینکه وارد حساب کاربری شده‌اند یا خیر، به آن دسترسی دارند، قسمت دوم (فایل userRouter) مسیرهایی است که کاربر حتما باید وارد حساب کاربری‌اش شده باشد و مسیر سوم (فایل adminRouter) مسیرهایی است که فقط کاربر ادمین به آن دسترسی دارد؛ نکته مهم در این قسمت این است که کاربران سطح بالاتر به مسیرهای سطح پایین‌تر نیز دسترسی دارند یعنی همانطور که بالاتر اشاره شد، کاربری که وارد حساب کاربری شده به مسیرهای کاربر آزاد (مهمان) دسترسی دارد و کاربر ادمین به تمامی مسیرها دسترسی دارد.

```
var indexRouter = require('./routes/index');
var usersRouter = require('./routes/users');
var adminRouter = require('./routes/admin');

const { ensureAuthenticated, ensureAdminAuthenticated } = require('./config/auth');
//***** <<Routes>> *****//
app.use('/', indexRouter);
app.use('/admin', ensureAdminAuthenticated, adminRouter);
app.use('/', ensureAuthenticated, usersRouter);
```

تصویر ۲۱

پس از اینکه کدهای مسیر دهی را به سه قسمت تقسیم کردیم لازم است قسمتی نیز برای بررسی سطح دسترسی کاربر تعبیه کنیم که ما در این برنامه این قسمت را به عنوان یک میدلور (میان افزار) تعریف کرده ایم که خروجی آن یا true و یا false است.

یک میان افزار به نام ensureAuthenticated برای بررسی اینکه آیا کاربر وارد حساب کاربری اش شده است یا خیر و یک میان افزار به نام ensureAdminAuthenticated برای اینکه آیا کاربر ادمین است یا خیر.

```
module.exports = {
  ensureAuthenticated: function (req, res, next) {
    if(req.isAuthenticated()) {
      return next();
    }
    req.flash('error_msg', 'لطفا وارد شوید');
    res.redirect('/login');
  },
  ensureAdminAuthenticated: function(req, res, next) {
    if(req.isAuthenticated() && req.user.userName === "Administrator") {
      return next();
    }
    req.flash('error_msg', 'رمز یا نام کاربری اشتباه است');
    res.redirect('/adminLogin');
  }
};
```

## تصویر ۲۲

در صورتی که کاربر خواسته یا ناخواسته وارد قسمت هایی شود که نیازمند دسترسی سطح بالاتر باشد وارد قسمت ورود کاربر شده و پیام "لطفا وارد شوید" برای او به نمایش در خواهد آمد.

## فصل ۵

### تشریح سایت سفرنامه






The login form is centered on a dark background. It features a title 'ورود' (Login) with a right-pointing arrow icon. Below the title are two input fields: the first is labeled 'رایانامه' (Email) and the second is labeled 'رمز' (Password). Both fields have placeholder text indicating where to enter the respective information. A blue button labeled 'ورود' (Login) is positioned below the password field. At the bottom of the form, there is a green link that says 'حساب ندارید؟ هم اکنون ثبت نام کنید' (Don't have an account? Register now).

تصویر ۲۳

در این قسمت کاربر با وارد کردن رایانامه و رمز خود، وارد حساب خود می‌شود، همچنین در صورتی که حساب ندارد به راحتی می‌تواند با فشردن قسمت "هم اکنون ثبت نام کنید"، اقدام به ایجاد یک حساب کاربری جدید کند. همچنین در صورتی که کاربر رمز خود را اشتباه وارد کرده باشد با پیام زیر روبرو می‌شود:



The error message is displayed in a dark orange banner with a white border. It contains the text 'رمز اشتباه است' (Password is incorrect) in white, followed by a white 'X' icon in the top right corner.

تصویر ۲۴

در صورتی که کاربر، آدرس رایانامه را اشتباه وارد کرده باشد با پیام زیر روبرو می‌شود:

آدرس رایانامه اشتباه است



تصویر ۲۵

ثبت نام

## ثبت نام

عکس کاربری

Browse

نام کاربری

نام را وارد کنید

رایانامه

رایانامه را وارد کنید

رمز عبور

رمز را وارد کنید

تکرار رمز عبور

رمز را مجددا وارد کنید

از خودت بنویس

من کی‌م؟

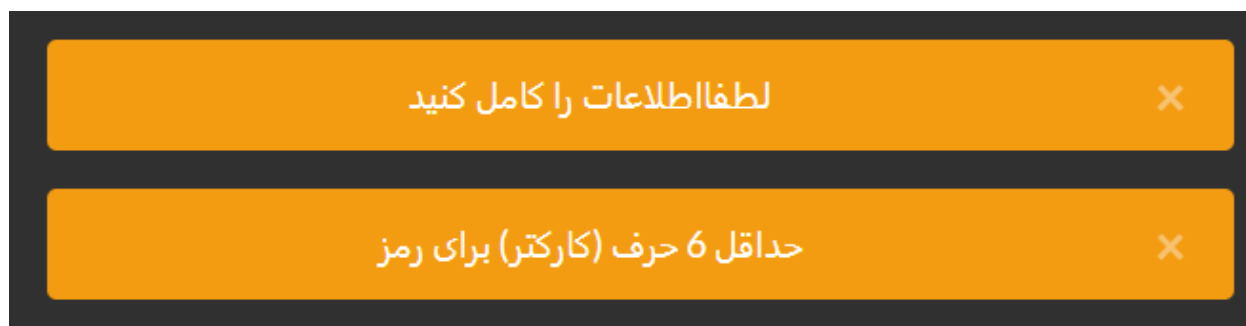
ثبت نام

حساب کاربری دارید؟ ورود

تصویر ۲۶

پس از آن که کاربر اطلاعات مورد نیاز را تکمیل کرد، کافیسیت کلید ثبت نام را فشار دهد.

همچنین برنامه تمهیداتی را برای عدم تکمیل اطلاعات توسط کاربر اندیشیده، که در ادامه به ذکر چند نمونه از آن بسنده می‌کنیم:



تصویر ۲۷

در صورتی که کاربر تکرار رمز را صحیح وارد نکند:



تصویر ۲۸

در صورتی ثبت نام موفقیت آمیز باشد به صفحه ورود منتقل می‌شود.

ورود

شما ثبت نام کردید و میتوانید وارد شوید

رایانامه

رایانامه را وارد کنید

رمز

رمز را وارد کنید

ورود

حساب ندارید؟ هم اکنون ثبت نام کنید

تصویر ۲۹

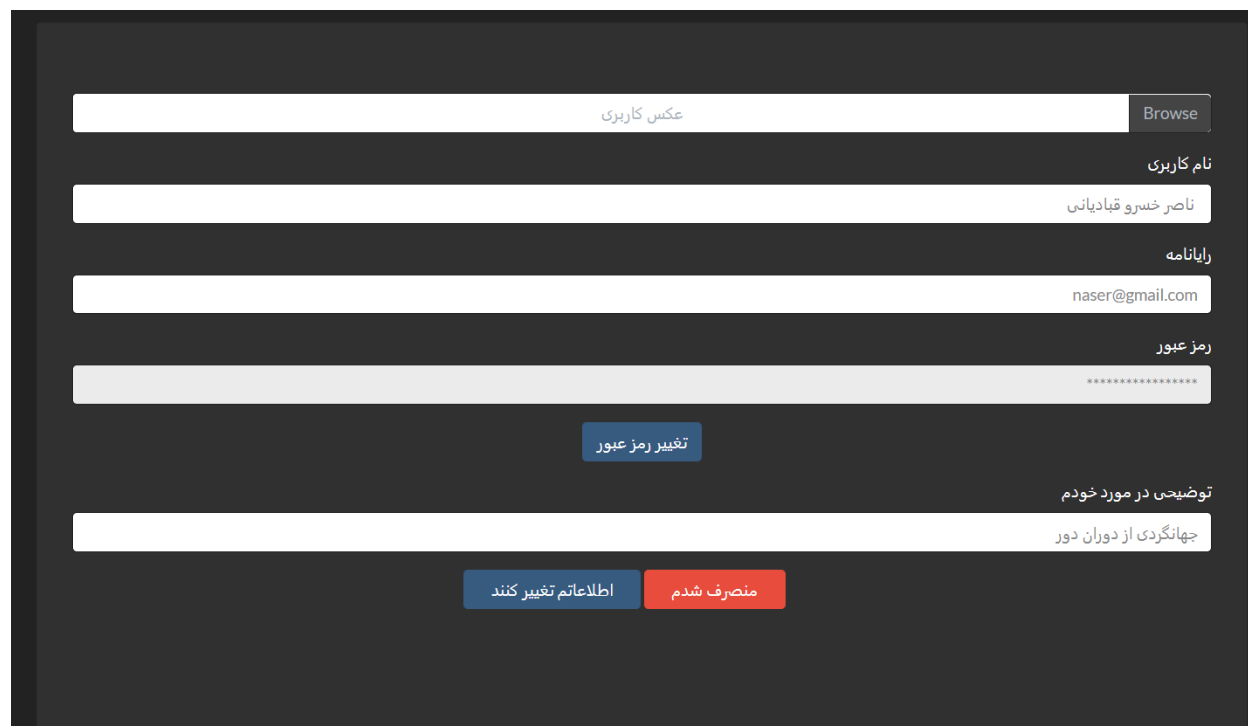
همچنین در پایگاه داده یک کاربر جدید ایجاد می‌شود:

(1) 602034826156ca16c82aa221 {10 fields}		Document
_id	602034826156ca16c82aa221	ObjectId
profileImage	/images/profileImages/defaultImage/profile.png	String
description	جهانگردی از دوران کهن	String
userName	نام‌ن کاربر قدیمی	String
email	naser@gmail.com	String
password	\$2a\$10\$DY9ur5c4o7u.ifsWKQip7ukX2NlgVc97yUthVQN7EdU8birmn7tfO	String
lastActivity	2/7/2021, 10:12:10 PM	Date
createdAt	2/7/2021, 10:12:10 PM	Date
updatedAt	2/7/2021, 10:18:37 PM	Date
__v	0	Int32

تصویر ۳۰

لازم به ذکر است از نرم افزار NoSQLBooster برای مدیریت پایگاه داده استفاده می کنیم.

## ویرایش اطلاعات کاربری



تصویر ۳۱

همچنین در صورتی که کاربر بخواهد رمز عبور خود را تغییر دهد، کافیست روی دکمه تغییر رمز عبور بفشارد تا قسمت تغییر رمز عبور را مشاهده کند:

عکس کاربری

Browse

نام کاربری

ناصر خسرو قبادیانی

رایانامه

naser@gmail.com

رمز قدیمی

\*\*\*\*\*

رمز جدید

\*\*\*\*\*

تکرار رمز جدید

\*\*\*\*\*

توضیحی در مورد خودم

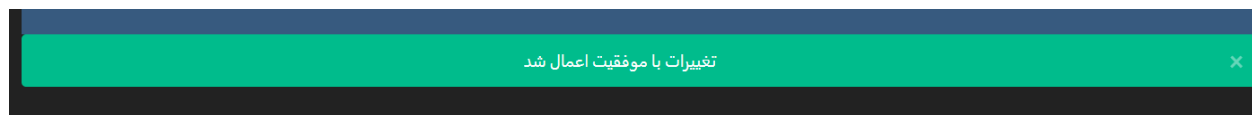
جهانگردی از دوران دور

اطلاعاتم تغییر کنند

منصرف شدم

تصویر ۳۲

در صورتی که اطلاعات با موفقیت تغییر کند، کاربر با پیام زیر مواجه می‌شود:



تصویر ۳۳

## افزودن یک خاطره

عکس مقاله

Browse

موضوع\*

اسم چی بذاریم

خلاصه‌ای از مقاله‌ی خود بنویسید\*

برگی از مقاله‌ام

مقاله را در این قسمت بنویسید\*

می‌نویسم هرآنچه را که از دلم فوران می‌کند

کلید واژه های مقاله خود را بنویسید

کلید واژه‌ها را با فاصله از هم جدا کنید.

☐ در کانال تلگرام سفرنامه منتشر کنیم؟

به خاطراتمون اضافه کنیم

منصرف شدم

### تصویر ۳۴

برای افزودن خاطره کافیت قسمت‌هایی که با ستاره مشخص شده‌اند تکمیل شود، بارگذاری عکس خاطره به صورت انتخابی به عهده کاربر است و برنامه در صورتی که کاربر عکسی بارگذاری نکند، یک عکس به عنوان پیش فرض در نظر گرفته است. همچنین انتشار خبر خاطره جدید در کانال تلگرام نیز برای کاربر در نظر گرفته شده که در صورت تمایل کاربر کافیت که آن را فعال کند. و در نهایت بعد از آنکه کاربر بر روی دکمه "به خاطراتمون اضافه کنیم" فشرده، خاطره (پست) جدید در پایگاه داده ذخیره می‌شود.

Key	Value	Type
(1) 6026109320633c09a47db1af	{ subject : "سفری جذاب به دکن آندریچان" } (10 fields)	Document
_id	6026109320633c09a47db1af	ObjectId
image	/images/postImages/defaultImage/defaultImage.jpg	String
articleKeys	[ "بکر" ]	Array
subject	سفری جذاب به دکن آندریچان	String
summery	در ترمینال بانک. همنفر می پرسد یا کدام خط باید برویم مرکز شهر؟ گاهی به یادداشت هیم می اندازم و شماره سه خطی که به مرکز شهر می روند را می گویم	String
article	وارد می ش ... آرام است و گل مسیر را می توانیم بخوابیم. 6 صبح است که به تهران می رسیم و با کوله باری از خاطرات یک سفر دیگر. راهی خانه می شویم	String
author	60260f5420633c09a47db1ae	ObjectId
createdAt	2/12/2021, 8:52:27 AM	Date
updatedAt	2/12/2021, 8:52:27 AM	Date
_v	0	Int32

## تصویر ۳۵

## پسندیدن یک خاطره

در صورتی که کاربر وارد حساب کاربری خود نشده باشد تنها قادر به مشاهده تعداد نفراتی که خاطره مورد نظرش را پسندیده‌اند، است.



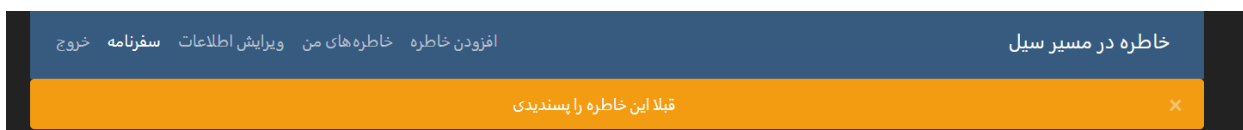
## تصویر ۳۶

در صورتی که وارد حساب کاربری خود شده باشد یک آیکون پسندیدن برای او به نمایش در خواهد آمد که در صورت علاقه در این حالت امکان پسندیدن خاطره (پست) مورد نظرش برای او فراهم شده است.



## تصویر ۳۷

همچنین در صورتی که یک بار قبلا پسندیده باشد امکان دوباره برای او وجود نخواهد داشت و در صورت فشردن دوباره این آیکون با پیام زیر مواجه می‌شود.





## منابع

- [/https://www.cyberpolice.ir](https://www.cyberpolice.ir)
- [/https://^learn.com](https://^learn.com)
- [/https://rocket.ir](https://rocket.ir)
- [/https://www.roxo.ir](https://www.roxo.ir)
- [/https://nodejs.org](https://nodejs.org)
- [/https://medium.com](https://medium.com)
- [/https://www.npmjs.com](https://www.npmjs.com)
- [/https://padidelearn.com](https://padidelearn.com)
- [/https://adminsite.com](https://adminsite.com)
- [/https://yadify.com](https://yadify.com)
- [/https://www.webcade.ir](https://www.webcade.ir)
- سفرنامه ها | لست سکند (lastsecond.ir)

آدرس گیت هاب پروژه:

MahdiYasini/university-final-project: The final project of the bachelor's degree in computer engineering (github.com)