برنادا

نام و نام خانوادگی توسعه دهنده:ابوالفضل محسنی  
کد ملی:1051460301

**معرفی پروژه برنادا**

همه‌ی دانش‌آموزان دوست دارند وقت‌شان را بهتر مدیریت کنند، اما برنامه‌ریزی درست همیشه چالش‌برانگیز است. بسیاری از دانش‌آموزان نمی‌دانند چگونه مطالعه‌ی خود را به‌طور منظم و بهینه سازمان‌دهی کنند، و این موضوع باعث کاهش بهره‌وری و استرس بیشتر در زمان امتحانات می‌شود.

**برنادا** یک پلتفرم تحت وب است که برای حل این مشکل طراحی شده و به دانش‌آموزان کمک می‌کند تا با استفاده از اصول علمی برنامه‌ریزی، یک جدول مطالعه‌ داشته باشند. این پروژه که توسط **ابوالفضل محسنی** توسعه یافته است، با در نظر گرفتن نیازهای مختلف دانش‌آموزان، یک برنامه‌ی درسی متناسب با زمان و شرایط آن‌ها ارائه می‌دهد.

برنادا با هدف افزایش بهره‌وری دانش‌آموزان، کاهش سردرگمی در برنامه‌ریزی درسی طراحی شده است. این پلتفرم به کاربران این امکان را می‌دهد که ساعات مطالعه‌ی خود را مشخص کنند و برنامه‌ای متناسب با نیازهایشان دریافت کنند.  
**دلیل ایجاد پروژه**

ایده‌ی ایجاد **برنادا** از تجربه‌ی شخصی من به‌عنوان یک دانش‌آموز نشأت گرفت. همیشه در مدیریت زمان و برنامه‌ریزی درسی با چالش‌هایی روبه‌رو بودم و گاهی اوقات وقتی برنامه‌ای نمی‌ساختم، هیچ چیزی برای یادآوری و پیگیری پیشرفتم وجود نداشت. به همین دلیل، تصمیم گرفتم یک پلتفرم تحت وب طراحی کنم که به دانش‌آموزان کمک کند تا برنامه‌های درسی خود را به‌طور مؤثر و به‌راحتی سازمان‌دهی کنند.

با توجه به نیاز روزافزون دانش‌آموزان به ابزاری برای مدیریت زمان و برنامه‌ریزی دقیق، این پروژه به‌طور خاص برای فراهم کردن یک ابزار آنلاین و قابل دسترس طراحی شد تا دانش‌آموزان بتوانند به‌سادگی برنامه‌های روزانه خود را تنظیم کنند و همیشه یک مرجع قابل مشاهده برای پیگیری پیشرفت‌شان داشته باشند.  
بخش های مختلف برنادا

**روند ثبت‌نام و ورود**

برای استفاده از پلتفرم **برنادا**، کاربران ابتدا باید در سیستم ثبت‌نام کنند. برای ثبت‌نام، کاربران نیاز به وارد کردن اطلاعات زیر دارند:

1. **نام**
2. **نام خانوادگی**
3. **شماره موبایل معتبر**
4. **رمز عبور** که باید حداقل 8 کاراکتر باشد و ترکیبی از اعداد و حروف انگلیسی باشد.

پس از ثبت‌نام، کاربر می‌تواند با وارد کردن **شماره تلفن** و **رمز عبور** خود در فرم ورود، به حساب کاربری خود دسترسی پیدا کند و از امکانات مختلف پلتفرم استفاده کند.

در حال حاضر، هیچ ویژگی برای تغییر اطلاعات یا بازیابی رمز عبور در نظر گرفته نشده است.  
  
**نمای کلی**

در بخش **نمای کلی**، کاربر می‌تواند به‌طور واضح لیست **کارهای امروز** خود و **کارهای عقب‌افتاده** را مشاهده کند. این بخش به‌عنوان یک نمای کلی از تمام فعالیت‌های پیش‌رو و گذشته به کاربر ارائه می‌شود تا بتواند زمان خود را بهتر مدیریت کند.

* **کارهای امروز**: این لیست شامل تمام فعالیت‌هایی است که کاربر برای امروز برنامه‌ریزی کرده است. هر زمان که کاربر یک فعالیت را به اتمام رساند، می‌تواند آن را با زدن یک **تیک** علامت‌گذاری کند تا نشان دهد که کار مورد نظر را انجام داده است.
* **کارهای عقب‌افتاده**: در این قسمت، کارهایی که کاربر نتوانسته است در روزهای گذشته انجام دهد، لیست می‌شود.

این ویژگی به کاربر این امکان را می‌دهد که همیشه از وضعیت کارهای خود آگاه باشد و بتواند با دیدی واضح‌تر تصمیم‌گیری کند. همچنین، با استفاده از تیک زدن، کاربر می‌تواند به راحتی پیگیری کند که چه کارهایی تمام شده و چه کارهایی هنوز باقی‌مانده است.  
 **کارهای امروز**

در بخش **کارهای امروز**، تمام فعالیت‌هایی که برای امروز برنامه‌ریزی شده است در قالب یک **جدول** نمایش داده می‌شود. این جدول جزئیات دقیقی از هر فعالیت را شامل می‌شود، از جمله:

* **ساعت شروع**: زمان دقیق شروع هر فعالیت
* **ساعت پایان**: زمان دقیق پایان هر فعالیت
* **نام فعالیت**: شرح کار یا فعالیتی که باید انجام شود

این بخش به‌طور دقیق‌تری به برنامه‌ریزی امروز کاربر پرداخته و جزئیات بیشتری را نسبت به **نمای کلی** ارائه می‌دهد. برخلاف نمای کلی، در این بخش امکان تیک زدن فعالیت‌ها وجود ندارد، و کاربر باید از بخش **نمای کلی** برای علامت‌گذاری انجام شده‌ها استفاده کند. این ساختار باعث می‌شود که کاربر به‌طور دقیق‌تر و به‌صورت زمان‌بندی‌شده بتواند فعالیت‌های خود را مشاهده کند و طبق برنامه پیش برود.  
  
**کارهای عقب‌افتاده**

بخش **کارهای عقب‌افتاده** مشابه بخش **کارهای امروز** است، با این تفاوت که اینجا کارهایی که کاربر نتوانسته در روزهای گذشته انجام دهد، نمایش داده می‌شود. در این بخش، برای هر کار، جزئیات مشابه با **کارهای امروز** ارائه می‌شود، از جمله:

* **ساعت شروع**: زمان دقیق شروع فعالیت
* **ساعت پایان**: زمان دقیق پایان فعالیت
* **نام فعالیت**: شرح کاری که باید انجام شود

همچنین، در این بخش هم مانند بخش **کارهای امروز**، **امکان تیک زدن وجود ندارد**. کاربر باید از بخش **نمای کلی** برای علامت‌گذاری کارهای انجام‌شده استفاده کند. هدف از این بخش این است که کاربر بتواند به راحتی کارهای عقب‌افتاده‌ی خود را شناسایی کرده و برای برنامه‌ریزی روزهای آینده آن‌ها را در نظر بگیرد.  
 **کل برنامه**

بخش **کل برنامه** به کاربر این امکان را می‌دهد که **تمام فعالیت‌های برنامه‌ریزی شده** را در یک **جدول منظم** مشاهده کند. این جدول به‌طور کامل تمام اطلاعات مورد نیاز برای هر فعالیت را نمایش می‌دهد، که شامل:

* **نام فعالیت**: شرح کار یا فعالیتی که باید انجام شود
* **ساعت شروع**: زمان دقیق شروع فعالیت
* **ساعت پایان**: زمان دقیق پایان فعالیت
* **روز هفته**: روزی که فعالیت در آن روز انجام می‌شود

این بخش به‌عنوان یک نمای کلی از تمام برنامه‌های کاری در طول هفته عمل می‌کند و به کاربر کمک می‌کند تا از برنامه‌ریزی دقیق‌تری برای روزهای مختلف هفته بهره‌مند شود. با این جدول، کاربر می‌تواند فعالیت‌ها را به‌طور واضح مشاهده کرده و از آن‌ها برای مدیریت زمان بهتر استفاده کند.  
  
  
  
**ساخت برنامه**  
در بخش **ساخت برنامه**، کاربر پس از ورود به این بخش، فرم مخصوصی را پر می‌کند که شامل اطلاعات زیر است:

1. **مقطع تحصیلی**: کاربر باید مقطع تحصیلی خود را انتخاب کند. مقاطع موجود شامل:
   * **ابتدایی**
   * **متوسطه اول**
   * **متوسطه دوم**
2. **پایه تحصیلی**: بسته به مقطع تحصیلی انتخاب شده، پایه تحصیلی متناسب با آن نمایش داده می‌شود. کاربر باید پایه تحصیلی خود را انتخاب کند.
3. **رشته تحصیلی** (فقط برای مقطع **متوسطه دوم**): اگر کاربر مقطع متوسطه دوم را انتخاب کند، می‌تواند یکی از رشته‌های تحصیلی خود را انتخاب کند. در نسخه ابتدایی **برنادا**، رشته‌های قابل انتخاب شامل:
   * **انسانی**
   * **تجربی**
   * **ریاضی**
4. **ساعت شروع برنامه**: کاربر می‌تواند ساعتی که می‌خواهد برنامه‌ریزی خود را آغاز کند، از هر ساعت از شبانه‌روز انتخاب کند.
5. **ساعات مطالعه**: کاربر باید انتخاب کند که روزانه چند ساعت می‌خواهد مطالعه کند. انتخاب‌ها شامل:
   * **2 ساعت**
   * **4 ساعت**
   * **6 ساعت**

پس از تکمیل فرم و زدن دکمه **ساخت برنامه**، برنامه‌ریزی درسی متناسب با اطلاعات وارد شده برای کاربر ساخته می‌شود و کاربر می‌تواند به برنامه‌ی خود دسترسی داشته باشد و آن را برای   
روزهای آینده دنبال کند.  
**راهنما**

بخش **راهنما** در صفحه اصلی **برنادا** به‌طور ویژه برای کاربرانی طراحی شده است که نیاز به آگاهی بیشتر در مورد نحوه استفاده از پلتفرم دارند. این بخش به‌طور **عمومی** در دسترس است و کاربر بدون نیاز به ورود به حساب کاربری می‌تواند به آن دسترسی پیدا کند. در این بخش، اطلاعات کاملی از تمام بخش‌ها و قابلیت‌های پلتفرم آورده شده است، که شامل موارد زیر است:

1. **ثبت‌نام و ورود**: توضیحات کامل در مورد روند **ثبت‌نام**، **ورود** به حساب کاربری و جزئیات فرم‌هایی که باید تکمیل شوند، همچنین توضیحاتی در مورد نیاز به یک **شماره موبایل معتبر** و نحوه ورود با استفاده از **شماره تلفن** و **رمز عبور**.
2. **ساخت برنامه**: راهنمای گام به گام برای پر کردن فرم **ساخت برنامه**، شامل انتخاب مقطع تحصیلی، پایه تحصیلی، رشته تحصیلی، ساعت شروع برنامه و ساعات مطالعه. همچنین، نحوه ساخت برنامه و دسترسی به آن پس از تکمیل فرم توضیح داده می‌شود.
3. **کارهای امروز و کارهای عقب‌افتاده**: توضیحاتی در مورد اینکه چگونه کاربر می‌تواند کارهای روزانه خود را مشاهده کند، از جمله **ساعت شروع و پایان هر کار** و امکان مشاهده کارهای عقب‌افتاده که باید در آینده انجام شوند.
4. **نمای کلی و کل برنامه**: توضیحات در مورد بخش‌های **نمای کلی** و **کل برنامه** که به کاربر کمک می‌کند تا تمام فعالیت‌های خود را به صورت واضح و سازمان‌یافته مشاهده کند.

این بخش به‌عنوان یک منبع آموزشی برای کاربرانی که تازه با پلتفرم آشنا می‌شوند و نیاز به راهنمایی دارند، عمل می‌کند. تمام توضیحات به‌طور ساده و شفاف در دسترس است و کاربر می‌تواند به راحتی با بخش‌های مختلف سیستم آشنا شود.

**ارتباط با ما**

بخش **"ارتباط با ما"** در **برنادا** به کاربران این امکان را می‌دهد تا در صورت نیاز به راهنمایی، پشتیبانی، یا ارسال بازخورد، با تیم توسعه در ارتباط باشند.

در این بخش، **دو روش ارتباطی** برای کاربران در نظر گرفته شده است:

1. **ایمیل و شماره تلفن**: کاربران می‌توانند از طریق اطلاعات تماس درج‌شده، مستقیماً با تیم پشتیبانی ارتباط بگیرند.
2. **فرم ارسال پیام**:

* کاربران می‌توانند با **وارد کردن شماره تماس، انتخاب موضوع پیام، و نوشتن توضیحات**، درخواست یا مشکل خود را ارسال کنند.
* این پیام‌ها از طریق **سرویس EmailJS** مستقیماً به **صندوق ورودی ایمیل برنادا** ارسال می‌شوند.

**هدف از این بخش**، فراهم کردن یک راه ارتباطی سریع و کارآمد برای کاربران است تا بتوانند مشکلات خود را مطرح کنند یا نظراتشان را با تیم توسعه به اشتراک بگذارند.  
 **ساختار کلی پروژه برنادا**

پروژه **برنادا** به‌صورت **ساختارمند و ماژولار** طراحی شده است تا توسعه، نگهداری و گسترش آن ساده‌تر باشد. در ادامه، **شرح کلی بخش‌های مختلف پروژه** را داریم:  
**1.پوشه asset(فایل های استایل و اسکریپت فرانت اند)**

در این پوشه، **فایل‌های CSS و JavaScript** مربوط به هر بخش از پروژه ذخیره شده‌اند.

* هر بخش از سایت دارای یک پوشه‌ی جداگانه است که نام آن **با نام پوشه‌ی متناظر در views/ یکسان است**.
* این ساختار باعث **نظم بهتر و دسترسی سریع‌تر** به فایل‌های مرتبط با هر صفحه شده است.  
  **2.پوشه views(فایل های HTML پروژه)**
* تمامی **فایل‌های HTML** در این پوشه قرار دارند.
* ساختار داخلی این پوشه شامل چندین **زیرپوشه** است که هرکدام مربوط به یک بخش از سایت می‌باشند:
* **landing/** → شامل صفحه‌ی اصلی سایت، درباره‌ی ما، راهنما و لینک‌ها.
* **LoginRegister/** → صفحات ورود و ثبت‌نام.
* **Dashboard/** → شامل صفحات مختلف داشبورد مانند برنامه امروز، کارهای عقب‌افتاده، ساخت برنامه، کل برنامه و نمای کلی.
* این ساختار باعث جداسازی بخش‌های مختلف سایت و مدیریت راحت‌تر آن شده است.

**3.پوشه routes(مدیریت روت های سمت سرور)**

این پوشه شامل **فایل‌های مربوط به مسیرهای مختلف API و مدیریت درخواست‌های سمت سرور** است.

* **server.js** → فایل اصلی سرور که تنظیمات اولیه و راه‌اندازی سرور را انجام می‌دهد.
* **auth.js** → مدیریت **ورود و ثبت‌نام کاربران**.
* **agenda.js** → ریست کردن وظایف در اول هر هفته.
* **getSchedule.js** → ارسال **برنامه درسی کاربر**.
* **task.js** → مدیریت تغییر وضعیت انجام برنامه ها.
* **schedule.js** → ساخت و مدیریت **برنامه درسی کاربران**.

4.پوشه models(مدل های دیتابیس)

* در این پوشه، مدل‌های **MongoDB** پروژه قرار دارند.
* **User.js** → مدل مربوط به کاربران که اطلاعات آن‌ها مانند **نام، شماره موبایل، رمز عبور** و **برنامه‌ی درسی** ذخیره می‌شود.
* **Schedule.js** → مدل مربوط به **برنامه‌های درسی کاربران** که شامل **مقطع تحصیلی، پایه، رشته، ساعات مطالعه، و زمان‌بندی کلی برنامه** است.
* **Task.js** → مدل مربوط به **فعالیت‌های روزانه** کاربران که شامل **نام فعالیت، ساعت شروع و پایان، روز انجام و وضعیت انجام‌شده یا انجام‌نشده** است.

**5.پوشه‌های تنظیمات و استایل اضافی**

* **setting/** → شامل فایل‌های **استایل تنظیمات** داشبورد.
* **Quick-access/** → شامل استایل های تکراری در بین صفحات.
* **link/** → استایل مربوط به صفحه‌ی لینک‌ها.
* **abuteMe/** → استایل صفحه‌ی درباره‌ی ما.

**6.فایل‌های مهم در ریشه‌ی پروژه**

* **package.json و package-lock.json** → شامل لیست پکیج‌های موردنیاز پروژه و تنظیمات مربوط به نود  
  **7.فایل‌های تنظیمات سرور و استقرار**
* در ریشه‌ی پروژه، فایل‌های مربوط به **تنظیمات سرور و استقرار** قرار دارند:
* **liara.json** → فایل پیکربندی برای **استقرار پروژه روی پلتفرم Liara** که شامل اطلاعاتی مانند نوع پلتفرم (Node.js)، متغیرهای محیطی، و تنظیمات اجرا است.
* **procfile** → فایل مربوط به **اجرای سرور در هاستینگ** که مشخص می‌کند هنگام استارت پروژه، چه دستوری اجرا شود .

**اجرای پروژه**

پروژه‌ی برنادا برای اجرا روی سرور تنظیم شده و APIهای آن بر روی دامنه‌ی https://bernada.ir پیکربندی شده‌اند. بنابراین، در صورت اجرای پروژه به‌صورت لوکال، ارتباط با دیتابیس برقرار نخواهد شد.

**۱. مشاهده‌ی نسخه‌ی آنلاین**

پروژه‌ی برنادا هم‌اکنون روی سرور **Liara** میزبانی شده و از طریق دامنه‌ی زیر در دسترس است:  
🔗 [**bernada.ir**](https://bernada.ir)

**۲. اجرای پروژه روی Liara**

برای اجرای پروژه روی لیارا، کافی است فایل‌های پروژه را روی این سرویس مستقر کرده و متغیرهای محیطی (.env) را از طریق پنل تنظیمات سرور مقداردهی کنید.

**۳. اجرای پروژه به‌صورت لوکال**

اگر قصد اجرای پروژه روی سیستم خود را دارید، این مراحل را دنبال کنید:

1. ابتدا **Node.js** را روی سیستم خود نصب کنید.
2. برای اجرای پروژه، دستور زیر را اجرا کنید:

npm start

با این کار، سرور به‌صورت لوکال راه‌اندازی می‌شود. اما **به دلیل عدم وجود اطلاعات دیتابیس در محیط لوکال، ارتباط با سرور و ذخیره‌سازی داده‌ها امکان‌پذیر نخواهد بود.**

**منابع استفاده‌شده**

**Frontend**

* **زبان‌ها:** HTML, CSS, JavaScript
* **کتابخانه‌ها و ابزارها:**
  + jQuery → برای تعاملات کاربری
  + EmailJS → برای ارسال پیام‌های ارتباط با ما
  + SweetAlert → برای نمایش پیام‌های هشدار و اعلان‌ها به شکل زیبا
  + Bootstrap Icons → مجموعه آیکون‌های سبک برای استفاده در رابط کاربری

**Backend**

* **زبان‌ها و فریمورک‌ها:**
  + Node.js و Express.js برای توسعه‌ی سرور
  + MongoDB برای مدیریت پایگاه داده
* **وابستگی‌های Backend:**
  + agenda → زمان‌بندی و مدیریت وظایف
  + bcrypt → هش کردن و امنیت رمزهای عبور
  + body-parser → پردازش داده‌های ارسال‌شده در درخواست‌ها
  + cors → مدیریت درخواست‌های کراس-اورجین
  + dotenv → مدیریت متغیرهای محیطی
  + express → فریمورک برای ساخت API
  + mongoose → مدیریت ارتباط با پایگاه داده MongoDB
  + nodemon → اجرا و ری‌استارت خودکار سرور در حین توسعه  
    **سرویس‌های خارجی**
  + **Liara** → برای میزبانی و اجرای سرور
  + **EmailJS** → برای ارسال پیام‌ها از فرم ارتباط با ما
  + **Google Search Console** → برای ایندکس کردن سایت در موتور جستجو