|  |  |
| --- | --- |
| گزارش پروژه کارشناسی | وب اپلیکیشن انبارداری    Warehouse Management Web Application  نام و نام خانوادگی : سعیدنجد  استاد راهنما : دکتر رضا مرتضوی  شماره دانشجویی : 98121042  خرداد ماه 1403 |



تأییدیه صحت و اصالت نتایج

اینجانب سعید نجد به شماره دانشجویی 98121042 دانشجوی رشته مهندسی کامپیوتر نرم افزار، مقطع تحصیلی کارشناسی ، تأیید می نمایم که کلیه نتایج این پایاننامه/رساله حاصل کار اینجانب و بدون هر گونه دخل و تصرف است و موارد نسخه برداری شده از آثار دیگران را با ذکر کامل مشخصات منبع ذکر کرده ام. در صورت اثبات خالف مندرجات فوق، به تشخیص دانشگاه مطابق با ضوابط و مقررات حاکم (قانون حمایت از حقوق مؤلفان و مصنفان و قانون ترجمه و تکثیر کتب و نشریات و آثارصوتی، ضوابط و مقررات آموزشی، پژوهشی و انضباطی...) با اینجانب رفتار خواهد شد و حق هر گونه اعتراض در خصوص احقاقحقوق مکتسب و تشخیص و تعیین تخلف و مجازات را از خویش سلب می نمایم. در ضمن، مسئولیت هرگونه پاسخ گویی به اشخاص اعم از حقیقی و حقوقی و مراجع ذیصالح (اعم از اداری و قضایی) به عهده اینجانب خواهد بود و دانشگاه هیچگونه مسئولیتی دراین خصوص نخواهد داشت.

سعید نجد

تاریخ و امضا:

مجوز بهره برداری از پایانامه

بهره برداری از این پایان نامه در چارچوب مقررات کتابخانه و باتوجه به محدودیتی که توسط استاد راهنما به شرح زیر تعیین می شود، بلامانع است:

• بهره برداری از این پایاننامه/ رساله برای همگان بلامانع است.

• بهرهبرداری از این پایاننامه/ رساله با اخذ مجوز از استاد راهنما، بلامانع است.

• بهره برداری از این پایاننامه/ رساله تا تاریخ .................................... ممنوع است.

نام استاد راهنما :جناب آقای دکتر رضا مرتضوی

تاریخ:

چکیده

این Warehouse Management System (WMS)با هدف مدیریت بهینه موجودی و اطلاعات مشتریان طراحی شده است. این سیستم با استفاده از زبان برنامه‌نویسی PHP برای بخش بک‌اند و زبان‌های HTML، CSS و JavaScript برای بخش فرانت‌اند توسعه یافته است. این نرم‌افزار به کاربران امکان می‌دهد تا اطلاعات کاملی از مشتریان و محصولات خود را ذخیره و مدیریت کنند. همچنین، قابلیت درج و به‌روزرسانی مشتریان و محصولات جدید را فراهم کرده است، که این امر به بهبود روندهای انبارداری و مدیریت مشتریان کمک شایانی می‌کند.

از دیگر ویژگی‌های این نرم‌افزار می‌توان به رابط کاربری ساده و کاربرپسند آن اشاره کرد که با استفاده از HTML، CSS و JavaScript طراحی شده است. این رابط کاربری به گونه‌ای طراحی شده که کاربران به راحتی می‌توانند با آن کار کنند و بدون نیاز به دانش فنی عمیق، به تمام امکانات سیستم دسترسی داشته باشند. با استفاده از این نرم‌افزار، کسب‌وکارها می‌توانند فرآیندهای انبارداری خود را بهبود بخشند و اطلاعات مشتریان و محصولات را به شکل منظم و یکپارچه مدیریت کنند.

فهرست

فصل اول: مقدمه ..................................................................................................................................1

مقدمه ای بر موضوع پروژه و ضرورت انجام آن .....................................................................................1

اهداف پروژه ............................................................................................................................1

معرفی محصول نرم افزاری و قابلیت های کلی آن ..................................................................................2

فصل دوم: تحلیل و طراحی سیستم .................................................................................................2

مقدمه .......................................................................................................................2

شرح نیازمندی های کاربر و سیستم .....................................................................................2

نیازمندی های کاربر .......................................................................................................2

نیازمندی های سیستم ...................................................................................................2

مدلسازی فرایند ...........................................................................................................2

لایه ها ی مهندسی نرم افزار ............................................................................................2

نمودار یوزک یس ............................................................................................................2

نمودار حالت..................................................................................................................2

نمودارER.....................................................................................................................2

موجودیت....................................................................................................2................

صفت .........................................................................................................................2

عنواع صفات کلید ..........................................................................................................2

صفات مرکب .................................................................................................................2

صفت مبنا و مشتق .........................................................................................................2

ارتباط .........................................................................................................................2

نمودار ER پروژه...............................................................................................................2

معماری نرم افزاری ..........................................................................................................2

لایه نمایشی (فرانت اند) ....................................................................................................

لایه منطقی (بک اند) ....................................................................................................

لایه داده (پایگاه داده) ....................................................................................................

تعامالت بین لایه ها ....................................................................................................

مزایای معماری پیشنهادی....................................................................................................

-1 فصل اول: مقدمه

1/1 مقدمه ای بر موضوع پروژه و ضرورت انجام آن

در دنیای کسب‌وکار امروزی، مدیریت بهینه انبار و اطلاعات مشتریان نقش حیاتی در موفقیت و بهره‌وری سازمان‌ها ایفا می‌کند. با افزایش حجم داده‌ها و پیچیدگی‌های مربوط به مدیریت موجودی و روابط مشتریان، نیاز به راه‌حل‌های نرم‌افزاری پیشرفته و کارآمد بیش از پیش احساس می‌شود. پروژه طراحی و توسعه نرم‌افزار انبارداری با هدف پاسخگویی به این نیاز ضروری به وجود آمده است. این نرم‌افزار با استفاده از فناوری‌های مدرن وب، امکان مدیریت جامع و یکپارچه انبار و اطلاعات مشتریان را فراهم می‌سازد، که نتیجه آن کاهش خطاها، بهبود روندهای کاری و افزایش رضایت مشتریان خواهد بود.

ضرورت انجام این پروژه از آنجا ناشی می‌شود که بسیاری از کسب‌وکارها هنوز از روش‌های سنتی و دستی برای مدیریت انبار و اطلاعات مشتریان استفاده می‌کنند، که این روش‌ها نه تنها وقت‌گیر و پرهزینه هستند، بلکه احتمال وقوع خطاهای انسانی نیز در آن‌ها بسیار بالا است. با استفاده از یک نرم‌افزار انبارداری مدرن و کارآمد، می‌توان این مشکلات را به حداقل رساند و فرآیندهای انبارداری را بهبود بخشید. این نرم‌افزار با قابلیت‌های متعددی مانند ثبت و به‌روزرسانی اطلاعات مشتریان و محصولات، گزارش‌گیری دقیق و سریع، و رابط کاربری کاربرپسند، ابزاری قدرتمند برای مدیران و کارکنان انبار خواهد بود تا بهره‌وری خود را افزایش دهند و در بازار رقابتی امروز، عملکرد بهتری داشته باشند.

**1/2 اهداف پروژه**

اهداف اصلی پروژه طراحی و توسعه نرم‌افزار انبارداری عبارتند از:

1. **مدیریت بهینه موجودی انبار:**
   * ایجاد یک سیستم دقیق و کارآمد برای ردیابی و مدیریت موجودی انبار به منظور کاهش خطاها و افزایش دقت در فرآیندهای انبارداری.
   * امکان مشاهده و به‌روزرسانی لحظه‌ای موجودی محصولات، که به تصمیم‌گیری‌های بهتر در زمینه سفارشات و تأمین کالاها کمک می‌کند.
2. **مدیریت اطلاعات مشتریان:**
   * توسعه یک پایگاه داده جامع برای ذخیره و مدیریت اطلاعات مشتریان، شامل جزئیات تماس، تاریخچه خریدها و ترجیحات مشتریان.
   * فراهم کردن امکان ثبت و به‌روزرسانی سریع و آسان اطلاعات مشتریان به منظور بهبود خدمات مشتری و افزایش رضایت مشتریان.
3. **قابلیت درج و به‌روزرسانی محصولات و مشتریان جدید:**
   * فراهم کردن رابط کاربری ساده و کاربرپسند برای اضافه کردن محصولات و مشتریان جدید به سیستم، که باعث تسریع در عملیات انبارداری و مدیریت مشتریان می‌شود.
   * ارائه ابزارهایی برای به‌روزرسانی اطلاعات موجود، که به بهبود دقت داده‌ها و کاهش هزینه‌های مربوط به خطاهای انسانی کمک می‌کند.
4. **افزایش بهره‌وری و کاهش هزینه‌ها:**
   * کاهش زمان و منابع مورد نیاز برای مدیریت دستی انبار و اطلاعات مشتریان از طریق اتوماسیون فرآیندها.
   * بهبود هماهنگی و ارتباطات داخلی در سازمان از طریق یک سیستم یکپارچه و قابل اعتماد.

این اهداف به گونه‌ای طراحی شده‌اند که به بهبود کارایی و بهره‌وری سازمان‌ها کمک کرده و فرآیندهای انبارداری و مدیریت مشتریان را به صورت قابل توجهی ارتقا دهند.

**1/3 معرفی محصول نرم‌افزاری**

نرم‌افزار انبارداری ما یک راه‌حل جامع و پیشرفته برای مدیریت بهینه انبار و اطلاعات مشتریان است که با استفاده از فناوری‌های مدرن وب توسعه یافته است. این نرم‌افزار با استفاده از زبان برنامه‌نویسی PHP برای بخش بک‌اند و زبان‌های HTML، CSS و JavaScript برای بخش فرانت‌اند، یک سیستم قدرتمند و کاربرپسند را فراهم می‌کند. این نرم‌افزار به کسب‌وکارها کمک می‌کند تا فرآیندهای انبارداری خود را بهبود بخشیده و مدیریت اطلاعات مشتریان را به صورت دقیق و کارآمد انجام دهند.

**قابلیت‌های کلی نرم‌افزار**

1. **مدیریت موجودی انبار:**
   * ردیابی و مدیریت دقیق موجودی محصولات با قابلیت مشاهده و به‌روزرسانی لحظه‌ای.
   * هشدارهای خودکار برای محصولات کم‌موجودی و تاریخ انقضا.
2. **مدیریت اطلاعات مشتریان:**
   * ثبت و نگهداری اطلاعات کامل مشتریان شامل نام، آدرس، شماره تماس و تاریخچه خریدها.
   * جستجو و فیلتر پیشرفته برای دسترسی سریع به اطلاعات مشتریان.
3. **درج و به‌روزرسانی محصولات و مشتریان جدید:**
   * رابط کاربری ساده و کاربرپسند برای اضافه کردن و به‌روزرسانی اطلاعات محصولات و مشتریان.
   * قابلیت دسته‌بندی و برچسب‌گذاری محصولات برای مدیریت بهتر.
4. **گزارش‌گیری و تحلیل داده‌ها:**
   * تولید گزارش‌های جامع و سفارشی شامل گزارش‌های فروش، موجودی، و عملکرد مشتریان.
   * داشبورد مدیریتی با نمایش آمار و نمودارهای مختلف برای تحلیل بهتر داده‌ها.
5. **رابط کاربری کاربرپسند:**
   * طراحی واکنش‌گرا (Responsive Design) که امکان استفاده آسان در دستگاه‌های مختلف (رایانه، تبلت، موبایل) را فراهم می‌کند.
   * ناوبری ساده و دسترسی سریع به تمام قابلیت‌های سیستم.
6. **امنیت و پشتیبانی:**
   * استفاده از پروتکل‌های امنیتی برای حفاظت از داده‌های حساس مشتریان و انبار.
   * پشتیبانی فنی و به‌روزرسانی‌های دوره‌ای برای اطمینان از عملکرد بهینه نرم‌افزار.

این نرم‌افزار با ارائه قابلیت‌های گسترده و کارآمد، ابزار قدرتمندی برای کسب‌وکارها فراهم می‌کند تا بتوانند انبار و اطلاعات مشتریان خود را به صورت بهینه مدیریت کنند و بهره‌وری خود را افزایش دهند.

فصل دوم: تحلیل و طراحی سیستم

2/1 مقدمه

در این بخش به تحلیل و طراحی سیستم انبارداری می‌پردازیم. ابتدا در لایه تحلیل به بررسی اجمالی با دید کاربر می‌پردازیم سپس در لایه طراحی دید خود را نزدیک به دید ماشین می‌کنیم و در آخر در لایه پیاده‌سازی مراحل انجام کار را در پی می‌گیریم. در این فصل از سه نمودار use case و state برای فاز تحلیل و از ER و نمایش جدولی برای نمایش طراحی پایگاه داده استفاده شده است.

### 2.2 شرح نیازمندی‌های کاربر و سیستم

#### 2.2.1 نیازمندی های کاربر

**مدیریت موجودی انبار:**

* کاربران باید بتوانند به سادگی اطلاعات موجودی محصولات مختلف از جمله نام، تعداد، تاریخ ورود و خروج و محل نگهداری را مشاهده کنند.
* کاربران باید بتوانند موجودی هر محصول را به‌روزرسانی کنند و در صورت کاهش موجودی به زیر سطح مشخص، هشدار دریافت کنند.

**ثبت و به‌روزرسانی اطلاعات محصولات:**

* + کاربران باید امکان ثبت و به‌روزرسانی اطلاعات محصولات جدید را داشته باشند. این شامل فرم‌های ساده برای ورود داده‌ها و امکانات جستجو و فیلتر پیشرفته است.

**مدیریت اطلاعات مشتریان:**

* + کاربران باید بتوانند اطلاعات مشتریان را ثبت، به‌روزرسانی و مشاهده کنند. این اطلاعات شامل نام، آدرس، شماره تماس و تاریخچه خرید است.

**گزارش‌گیری و تحلیل:**

* + کاربران باید امکان دریافت گزارش‌های جامع از وضعیت انبار، فروش و عملکرد مشتریان را داشته باشند. این گزارش‌ها باید به صورت گرافیکی و قابل درک ارائه شوند.

#### 2.2.2 نیازمندی‌های سیستم

**مدیریت محتوا:**

* + مدیر سیستم باید بتواند محصولات و مشتریان را ایجاد، ویرایش و حذف کند.
  + مدیر سیستم باید بتواند موجودی محصولات را به‌روزرسانی کند و نظرات و درخواست‌های کاربران را مدیریت کند.

**رابط کاربری و تجربه کاربری:**

* + نرم‌افزار باید رابط کاربری ساده و کاربرپسند داشته باشد که از طریق آن کاربران به راحتی به اطلاعات محصولات و مشتریان دسترسی پیدا کنند.
  + طراحی واکنش‌گرا برای دسترسی بهتر از دستگاه‌های مختلف، از جمله موبایل و تبلت، ضروری است.

**پایداری و نگهداری:**

* + سیستم باید به گونه‌ای طراحی شود که به‌روزرسانی و نگهداری اطلاعات به سادگی قابل انجام باشد.
  + سیستم باید توانایی مقابله با بار ترافیکی بالا را داشته باشد.

**امنیت و حریم خصوصی:**

* + سیستم باید از داده‌های حساس محافظت کند.
  + باید از پروتکل‌های امنیتی مناسب برای حفاظت از اطلاعات کاربران استفاده شود.

### 2.3 مدلسازی فرآیند

#### 2.3.1 لایه‌های مهندسی نرم‌افزار

مهندسی نرم‌افزار یک فرآیند لایه‌ای است که از چهار لایه تشکیل شده است:

**کیفیت:**

* + مهندسی نرم‌افزار تلاش لایه‌ای برای تولید یک محصول نرم‌افزاری با کیفیت است که نیازمندی‌های مشتری را برآورده کند. این امر زمانی محقق می‌شود که ابزارها، روش‌ها و تکنیک‌های مناسب به درستی انتخاب و استفاده شوند.

**فرایندها:**

* + فرایند تولید نرم‌افزار شامل تبدیل نیازهای مشتری به یک محصول نرم‌افزاری است. این فرایند مشخص می‌کند که چه کسی، چه کاری، چه زمانی و چگونه باید انجام دهد تا محصولی با کیفیت تولید شود.

**روش‌ها:**

* + روش‌های ساخت نرم‌افزار به دو نوع "ساخت‌یافته" و "شیءگرا" تقسیم می‌شوند. در روش ساخت‌یافته، عملکردها و داده‌ها جداگانه تحلیل و طراحی می‌شوند، اما در روش شیءگرا، داده‌ها و عملکردها در قالب کلاس‌ها ترکیب می‌شوند.

**ابزارها:**

* + ابزارهای فنی برای ساخت نرم‌افزار به دو نوع، یعنی ابزار ساخت‌یافته و ابزار شیءگرا تقسیم می‌شوند.

#### 2.3.2 نمودار یوزکیس (Use Case)

Use Case ابزاری برای تعریف تعاملات مورد نیاز کاربر در سیستم انبارداری است. در واقع مجموعه اقداماتی است که مرحله به مرحله تعاملات بین کاربر و سیستم را برای رسیدن به یک هدف خاص تعریف می‌کند. در توسعه نرم‌افزار Use Cases یا مجموعه‌ای از Caseها نوشته می‌شود.

در ادامه نمودار Use Case پروژه انبارداری آمده است.

### 2.3.3 نمودار حالت

نمودار حالت نوعی دیاگرام برای تشریح و توضیح رفتار یک سیستم به صورت مرحله‌ای در علوم کامپیوتری و دیجیتال است. نمودار حالت، از تعداد محدودی «حالت» تشکیل شده‌است که این حالت‌ها بیانگر یکی از وضعیت‌های مختلف سیستم می‌باشد. برای یک نگاه کلی به سیستم، ابتدا نمودار حالت رسم شده و سپس از روی آن جدول حالت و شماتیک به دست می‌آید. این عمل به دلیل نزدیکی فهم بصری نمودار با کارکرد کلی سیستم می‌باشد. در ادامه، نمودار حالت پروژه انبارداری آمده است.

### 2.4 نمودار ER

نمودار (ER) Relation Entity اولین بار در سال 1976 توسط چن (Chen) از دانشگاه MIT جهت طراحی بانک‌های اطلاعاتی معرفی شد. این مدل به دلیل سادگی و قابلیت فهم بالا، اهمیت فراوانی پیدا کرد و به تدریج استفاده از آن گسترش یافت. در ادامه به مفاهیم موجود در ER می‌پردازیم.

#### 2.4.1 موجودیت

موجودیت‌ها پدیده‌هایی هستند که در بانک اطلاعاتی وجود خارجی دارند. برای مثال، در سیستم انبارداری موجودیت‌هایی همچون محصول، مشتری و سفارش وجود دارند. موجودیت‌ها دو نوع هستند:

* **مستقل:** موجودیت‌هایی که به خودی خود معنی دارند (مثال: محصول).
* **وابسته:** موجودیت‌هایی که وجودشان وابسته به موجودیت دیگر است (مثال: جزئیات سفارش که وابسته به سفارش است).

#### 2.4.2 صفت

هر موجودیت دارای ویژگی‌هایی است که آن را نسبت به موجودیت‌های دیگر متمایز می‌سازد، به این ویژگی‌ها صفت می‌گویند. برای مثال، برای موجودیت محصول، می‌توان صفاتی همچون کد محصول، نام، قیمت و تاریخ تولید تعریف کرد.

#### 2.4.3 صفات کلید

صفت کلید یک یا چند صفت از هر موجودیت است که مقادیر آن منحصر به فرد است. برای مثال، صفت کد محصول برای موجودیت محصول می‌تواند صفت کلید باشد. صفات کلید به دو نوع تقسیم می‌شوند:

* **کلید اصلی:** کلید اصلی موجودیت که با یک خط زیر نام صفت نمایش داده می‌شود.
* **کلید فرعی:** کلیدی که به سبب رابطه بین موجودیت‌ها به عنوان صفت این موجودیت در نظر گرفته شده و با دو خط زیر نام صفت مشخص می‌شود.

#### 2.4.4 صفات مرکب

برخی صفات ممکن است از ترکیب چند صفت کوچکتر به وجود آمده باشند. به چنین صفاتی صفات مرکب می‌گویند. برای مثال، صفت آدرس که می‌تواند به خیابان، کوچه و پلاک تقسیم شود. صفاتی که از بخش‌های کوچکتری تشکیل نشده باشند، صفات ساده نامیده می‌شوند.

#### 2.4.5 صفت مبنا و مشتق

صفات به دو دسته تقسیم می‌شوند:

* **مبنا:** صفاتی که مقادیر آنها به طور مستقیم در پایگاه داده ذخیره می‌شود (مثال: نام محصول).
* **مشتق:** صفاتی که مقادیر آنها در پایگاه داده ذخیره نشده و با توجه به مقادیر صفات دیگر تعیین می‌شود (مثال: سن که از تاریخ تولد محاسبه می‌شود). صفات مشتق با خط چین نمایش داده می‌شوند.

### معماری نرم‌افزاری پروژه انبارداری

معماری نرم‌افزاری این وب‌سایت شامل سه لایه اصلی است: لایه نمایشی (فرانت‌اند)، لایه منطقی (بک‌اند) و لایه داده (پایگاه داده). این سه لایه به صورت هماهنگ با یکدیگر کار می‌کنند تا وب‌سایت به درستی عملکرد داشته باشد.

#### 2.5.1 لایه نمایشی (فرانت‌اند)

* **HTML/CSS**: برای ساختاردهی و طراحی صفحات استفاده می‌شود.
* **JavaScript**: برای ایجاد تعاملات کاربری و پویا‌سازی صفحات به کار می‌رود.
* **Bootstrap**: برای ایجاد طراحی ریسپانسیو و زیبا‌سازی صفحات مورد استفاده قرار می‌گیرد.

#### 2.5.2 لایه منطقی (بک‌اند)

* **PHP**: به عنوان زبان برنامه‌نویسی اصلی برای توسعه بک‌اند استفاده می‌شود.
* **کنترلرها و مدل‌ها**: مدیریت منطق کسب و کار و ارتباط با پایگاه داده.

#### 2.5.3 لایه داده (پایگاه داده)

* **MySQL**: برای ذخیره‌سازی داده‌های مربوط به محصولات، مشتریان و تراکنش‌ها استفاده می‌شود. MySQL یک پایگاه داده رابطه‌ای است که به دلیل قابلیت اطمینان و کارایی بالا برای این نوع پروژه مناسب است.

#### 2.5.4 تعاملات بین لایه‌ها

* **فرانت‌اند با بک‌اند**: از طریق فرم‌ها و درخواست‌های HTTP که توسط PHP پردازش می‌شوند. فرانت‌اند درخواست‌هایی مانند افزودن مشتری جدید، نمایش لیست محصولات و ثبت تراکنش‌ها را به بک‌اند ارسال می‌کند.
* **بک‌اند با پایگاه داده**: با استفاده از MySQLi یا PDO برای اتصال به پایگاه داده و انجام عملیات CRUD (ایجاد، خواندن، به‌روزرسانی، حذف).

### مزایای معماری پیشنهادی

* **جداسازی مسئولیت‌ها**: هر لایه مسئول وظایف خاصی است که منجر به کاهش پیچیدگی و بهبود قابلیت نگهداری می‌شود.
* **مقیاس‌پذیری**: استفاده از PHP و MySQL امکان مقیاس‌پذیری بالا را فراهم می‌کند.
* **انعطاف‌پذیری**: تغییرات در یکی از لایه‌ها به سادگی امکان‌پذیر است بدون اینکه تاثیر زیادی بر سایر لایه‌ها بگذارد.
* **سرعت بارگذاری**: با استفاده از Bootstrap، صفحات به صورت ریسپانسیو و بهینه بارگذاری می‌شوند.

### فصل سوم: محیط و ابزار پیاده‌سازی

#### مقدمه

در این فصل، به معرفی محیط و ابزارهای پیاده‌سازی پروژه انبارداری پرداخته می‌شود. این بخش شامل توضیحاتی در مورد زبان‌های برنامه‌نویسی، فریمورک‌ها، کتابخانه‌ها و سایر ابزارهای مورد استفاده در پروژه می‌باشد. همچنین، محیط پیاده‌سازی شامل سیستم‌عامل‌ها، محیط‌های توسعه یکپارچه (IDE)، سرورها، ابزارهای تست و سایر ابزارهای جانبی به تفصیل شرح داده خواهد شد.

#### معرفی زبان‌های برنامه‌نویسی، فریمورک‌ها و کتابخانه‌های استفاده شده

##### 3.2 زبان‌های برنامه‌نویسی

* **PHP**: زبان اصلی مورد استفاده برای توسعه بک‌اند پروژه است.

##### 3.2.1 فریمورک‌ها و کتابخانه‌ها

* **PHP**: برای توسعه بخش بک‌اند استفاده می‌شود.
* **JavaScript**: برای ایجاد تعاملات و پویا‌سازی صفحات در فرانت‌اند استفاده می‌شود.
* **Bootstrap**: برای طراحی رابط کاربری ریسپانسیو و زیبا به کار می‌رود.
* **MySQLi**: برای ارتباط با پایگاه داده MySQL از این کتابخانه استفاده می‌شود.

##### 3.2.2 ابزارهای CSS

* **Bootstrap**: برای ایجاد طراحی ریسپانسیو و زیبا‌سازی صفحات.
* **HTML/CSS**: برای ساختاردهی و طراحی صفحات وب استفاده می‌شود.

#### شرح محیط پیاده‌سازی (سیستم‌عامل، IDE، ابزارهای جانبی)

##### 3.2.3 سیستم‌عامل‌ها

* **Windows 10**: سیستم‌عامل اصلی برای توسعه پروژه به دلیل پشتیبانی گسترده از ابزارهای توسعه و محیط کاربری مناسب.

##### 3.2.4 محیط‌های توسعه یکپارچه (IDE)

* **Visual Studio Code**: محیط توسعه یکپارچه (IDE) اصلی برای کدنویسی. VS Code به دلیل افزونه‌های متعدد و قابلیت‌های بالا برای توسعه وب بسیار محبوب است.
* **XAMPP**: برای اجرای سرور محلی و پایگاه داده MySQL.

##### 3.2.5 ابزارهای جانبی

* **Git**: سیستم کنترل نسخه برای مدیریت کد منبع و همکاری تیمی.
* **phpMyAdmin**: برای مدیریت پایگاه داده MySQL از طریق یک رابط کاربری وب.

### فصل چهارم: پیاده‌سازی و توسعه

#### مقدمه

در این فصل، به تفصیل فرآیند پیاده‌سازی و توسعه پروژه حسابداری با PHP شرح داده خواهد شد. این فصل شامل توضیحاتی در مورد ساختار پروژه و نحوه پوشه‌بندی آن، پیاده‌سازی پایگاه داده، طراحی و توسعه فرم‌ها و صفحات وب برای بخش‌های کاربری و مدیریت، و پیاده‌سازی عملیات‌های مختلف مانند ثبت، ویرایش، حذف و نمایش اطلاعات می‌باشد. هدف این بخش، ارائه یک دید جامع و دقیق از تمامی مراحل و جنبه‌های مختلف پیاده‌سازی پروژه به منظور درک بهتر فرآیندها و تصمیم‌گیری‌های فنی می‌باشد.

پروژه "WMS" شامل پوشه‌ها و فایل‌های زیر است که به منظور سازماندهی و پیاده‌سازی بهتر کدها طراحی شده‌اند:

1. **control.php**
   * فایل اصلی کنترل پروژه که وظایف مختلفی را مدیریت می‌کند.
2. **css/**
   * شامل فایل‌های CSS برای استایل‌دهی به صفحات وب.
3. **dist/**
   * پوشه‌ای برای فایل‌های تولید شده نهایی، معمولاً شامل کدهای کمینه‌سازی شده و آماده برای استقرار.
4. **fonts/**
   * شامل فونت‌های مورد استفاده در پروژه.
5. **images/**
   * شامل تصاویر مورد استفاده در وبسایت.
6. **js/**
   * شامل فایل‌های جاوااسکریپت برای تعاملات و عملکردهای دینامیک وبسایت.
7. **favicon.ico**
   * آیکون سایت.
8. **icon.png**
   * آیکون دیگر سایت، احتمالاً با رزولوشن بالاتر.
9. **index.php**
   * فایل اصلی ورود به سایت که معمولاً به عنوان صفحه اصلی سایت عمل می‌کند.

### توضیح پوشه‌بندی

* **control.php:** مدیریت و کنترل اصلی پروژه.
* **css/:** شامل فایل‌های CSS برای استایل‌دهی به صفحات.
* **dist/:** خروجی نهایی و کمینه‌سازی شده کدها برای استقرار.
* **fonts/:** فونت‌های مورد استفاده.
* **images/:** تصاویر مورد نیاز وبسایت.
* **js/:** اسکریپت‌های جاوااسکریپت برای تعاملات وبسایت.
* **favicon.ico و icon.png:** آیکون‌های سایت.
* **index.php:** نقطه ورود اصلی به وبسایت.

این ساختار به سازماندهی بهتر کدها کمک می‌کند و دسترسی و مدیریت آن‌ها را ساده‌تر می‌سازد.

### ساختار پایگاه‌داده

در پروژه " WMS "، ساختار پایگاه‌داده به‌گونه‌ای طراحی شده است که مدیریت داده‌ها به صورت کارآمد و منظم انجام شود. پایگاه‌داده به بخش‌های مختلفی تقسیم شده و هر بخش مسئول نگهداری از اطلاعات خاصی می‌باشد. این ساختار امکان گسترش و تغییرات آینده را نیز فراهم می‌کند.

### جزئیات پایگاه‌داده

پایگاه‌داده این پروژه شامل جداول و ارتباطات زیر است:

1. **جداول کاربران (users):**
   * نگهداری اطلاعات کاربرانی که به سیستم دسترسی دارند.
   * شامل فیلدهایی مانند شناسه کاربر، نام کاربری، رمز عبور، و نقش کاربری.
2. **جداول حسابداری (accounting):**
   * ثبت و مدیریت تراکنش‌های مالی و حسابداری.
   * شامل فیلدهایی مانند شناسه تراکنش، مبلغ، تاریخ تراکنش، نوع تراکنش و توضیحات.
3. **جداول محصولات (products):**
   * نگهداری اطلاعات محصولات مختلف.
   * شامل فیلدهایی مانند شناسه محصول، نام محصول، قیمت، و تعداد موجودی.
4. **جداول سفارشات (orders):**
   * مدیریت سفارشات ثبت شده توسط کاربران.
   * شامل فیلدهایی مانند شناسه سفارش، شناسه کاربر، شناسه محصول، تاریخ سفارش و وضعیت سفارش.

### ارتباطات بین جداول

پایگاه‌داده به گونه‌ای طراحی شده است که ارتباط منطقی و مناسبی بین جداول وجود داشته باشد. برای مثال:

* **ارتباط بین کاربران و سفارشات:** هر کاربر می‌تواند چندین سفارش داشته باشد.
* **ارتباط بین سفارشات و محصولات:** هر سفارش می‌تواند شامل چندین محصول باشد.

این طراحی به ما کمک می‌کند تا داده‌ها را به صورت منظم و کارآمد مدیریت کرده و گزارشات مالی دقیق و جامعی ارائه دهیم.

### تصاویر ساختار پایگاه‌داده

در ادامه، تصاویری از ساختار پایگاه‌داده پروژه آورده شده است که نشان‌دهنده جداول و ارتباطات بین آن‌ها می‌باشد:

استفاده از این ساختار، به ما این امکان را می‌دهد که پایگاه‌داده‌ای پایدار، قابل اطمینان و کارآمد داشته باشیم که به سادگی قابل گسترش و تغییر باشد و بتواند نیازهای مختلف پروژه را برآورده سازد.

### فصل پنجم: اجزا و عملکرد پروژه

#### مقدمه

در این فصل به معرفی و بررسی اجزای مختلف پیاده‌سازی شده در هر دو سمت فرانت‌اند و بک‌اند پروژه "Accounting with PHP" پرداخته می‌شود. همچنین عملکرد و بخش‌های کاربری و مدیریتی پروژه به تفصیل مورد بررسی قرار می‌گیرد.

#### پنل ادمین

##### ورود ادمین به پنل مدیریت

پنل مدیریت جایی است که مدیر سیستم می‌تواند به تمامی بخش‌های بک‌اند دسترسی داشته باشد. این پنل با استفاده از PHP پیاده‌سازی شده و امکانات مختلفی را برای مدیریت داده‌ها فراهم می‌کند.

##### صفحه اصلی پنل ادمین

صفحه اصلی پنل مدیریت شامل داشبوردی است که اطلاعات کلی درباره سیستم را نمایش می‌دهد. این صفحه به مدیر امکان می‌دهد تا به سرعت به بخش‌های مختلف سیستم دسترسی پیدا کند.

##### مدیریت کاربران

در این بخش، مدیر می‌تواند کاربران جدید اضافه کرده، اطلاعات کاربران موجود را ویرایش یا حذف کند. همچنین امکان تعیین سطوح دسترسی مختلف برای کاربران نیز وجود دارد.

##### مدیریت تراکنش‌ها

این بخش به مدیر امکان می‌دهد تا تراکنش‌های مالی را مدیریت کند. می‌توان تراکنش‌های جدید اضافه کرد، تراکنش‌های موجود را ویرایش یا حذف کرد و گزارش‌های مالی را مشاهده نمود.

#### بخش کاربری

##### داشبورد کاربری

داشبورد کاربری به کاربران امکان می‌دهد تا به اطلاعات مالی خود دسترسی داشته باشند. کاربران می‌توانند تراکنش‌های خود را مشاهده و مدیریت کنند.

##### مدیریت پروفایل

کاربران می‌توانند اطلاعات پروفایل خود را مشاهده و ویرایش کنند. این شامل اطلاعات شخصی، اطلاعات تماس و تنظیمات امنیتی می‌شود.

##### مشاهده و مدیریت تراکنش‌ها

کاربران می‌توانند تراکنش‌های مالی خود را مشاهده و مدیریت کنند. این بخش به کاربران امکان می‌دهد تا تراکنش‌های جدید اضافه کرده و تراکنش‌های موجود را ویرایش یا حذف کنند.

### فصل هفتم: نتیجه‌گیری و جمع‌بندی پروژه

#### مقدمه

در این فصل به جمع‌بندی و نتیجه‌گیری از پروژه "نرم‌افزار انبارداری" پرداخته خواهد شد. این بخش شامل بررسی دستاوردها، مشکلات و چالش‌ها، و پیشنهادات برای بهبود و توسعه آینده پروژه است.

#### دستاوردها

پروژه "نرم‌افزار انبارداری" با هدف ایجاد یک سیستم مدیریت انبار ساده و کارآمد پیاده‌سازی شد. دستاوردهای اصلی این پروژه شامل موارد زیر است:

* توسعه یک پنل مدیریت برای کنترل و نظارت بر موجودی انبار
* ایجاد بخش کاربری که به کاربران امکان مدیریت ورودی و خروجی کالاها را می‌دهد
* پیاده‌سازی امکانات امنیتی برای حفاظت از اطلاعات انبار
* استفاده از PHP و پایگاه داده MySQL برای ایجاد یک سیستم پایدار و کارآمد

#### مشکلات و چالش‌ها

در طول اجرای پروژه، با چالش‌ها و مشکلات مختلفی مواجه شدیم که برخی از مهم‌ترین آن‌ها عبارتند از:

* مشکلات مربوط به هماهنگی و یکپارچه‌سازی بخش‌های مختلف سیستم
* مدیریت و حفظ امنیت داده‌ها در یک محیط آنلاین
* بهینه‌سازی عملکرد سیستم برای اطمینان از سرعت و کارایی مناسب

#### پیشنهادات برای بهبود و توسعه آینده

برای بهبود و توسعه آینده پروژه، پیشنهادات زیر مطرح می‌شود:

* اضافه کردن امکانات پیشرفته‌تر برای تحلیل و گزارش‌گیری موجودی
* بهبود رابط کاربری برای افزایش کاربرپسندی و سهولت استفاده
* پیاده‌سازی نسخه‌های موبایل و وب‌اپلیکیشن برای دسترسی آسان‌تر کاربران
* بررسی و به‌روزرسانی مستمر امنیت سیستم برای جلوگیری از نفوذ و دسترسی غیرمجاز

#### نتیجه‌گیری

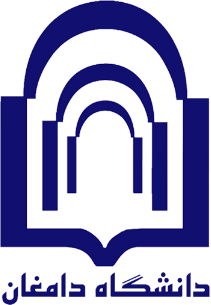
پروژه "نرم‌افزار انبارداری" با موفقیت به اهداف اولیه خود دست یافت و یک سیستم مدیریت انبار کارآمد و امن را ارائه داد. این پروژه می‌تواند به عنوان پایه‌ای برای توسعه‌های آتی و افزودن امکانات بیشتر مورد استفاده قرار گیرد. با توجه به چالش‌ها و مشکلات موجود، بهبود مستمر و بررسی دوره‌ای سیستم برای حفظ و ارتقای کیفیت آن ضروری است.

### مراجع

1. **PHP Official Documentation**
   * URL: <https://www.php.net/docs.php>
   * توضیحات: مستندات رسمی زبان برنامه‌نویسی PHP، شامل راهنماها، مثال‌ها و توضیحات جامع در مورد عملکردها و ویژگی‌های مختلف PHP.
2. **MySQL Official Documentation**
   * URL: <https://dev.mysql.com/doc/>
   * توضیحات: مستندات رسمی پایگاه داده MySQL، شامل راهنماهای نصب، پیکربندی، مدیریت و استفاده از MySQL.
3. **Bootstrap Official Documentation**
   * URL: https://getbootstrap.com/docs/
   * توضیحات: مستندات رسمی فریمورک Bootstrap، شامل راهنماها و مثال‌های استفاده از این فریمورک برای طراحی و توسعه رابط کاربری.
4. **JavaScript Official Documentation (MDN)**
   * URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
   * توضیحات: مستندات رسمی JavaScript در سایت MDN، شامل راهنماها و توضیحات جامع در مورد ویژگی‌ها و عملکردهای JavaScript.
5. **GitHub Repository of the Project**
   * URL: <https://github.com/saeednajd/acountingwithphp>
   * توضیحات: مخزن گیت‌هاب پروژه، شامل کد منبع، مستندات و فایل‌های مربوط به پروژه نرم‌افزار انبارداری.
6. **Visual Studio Code Documentation**
   * URL: <https://code.visualstudio.com/docs>
   * توضیحات: مستندات رسمی محیط توسعه یکپارچه Visual Studio Code، شامل راهنماها و توضیحات استفاده از این محیط توسعه برای کدنویسی و دیباگ.
7. **Composer Official Documentation**
   * URL: https://getcomposer.org/doc/
   * توضیحات: مستندات رسمی ابزار مدیریت وابستگی Composer برای PHP، شامل راهنماهای نصب و استفاده از Composer در پروژه‌های PHP.

این مراجع به عنوان منابع اصلی برای توسعه و پیاده‌سازی پروژه نرم‌افزار انبارداری مورد استفاده قرار گرفته‌اند و اطلاعات جامع و مفیدی در اختیار توسعه‌دهندگان قرار می‌دهند.

4o



School of Engineering Faculty of Computer Engineering

Bachelor project report

Warehouse Management Web Application

Saeed Najd

Supervisor

Dr. Reza Mortazavi

May - June 2024