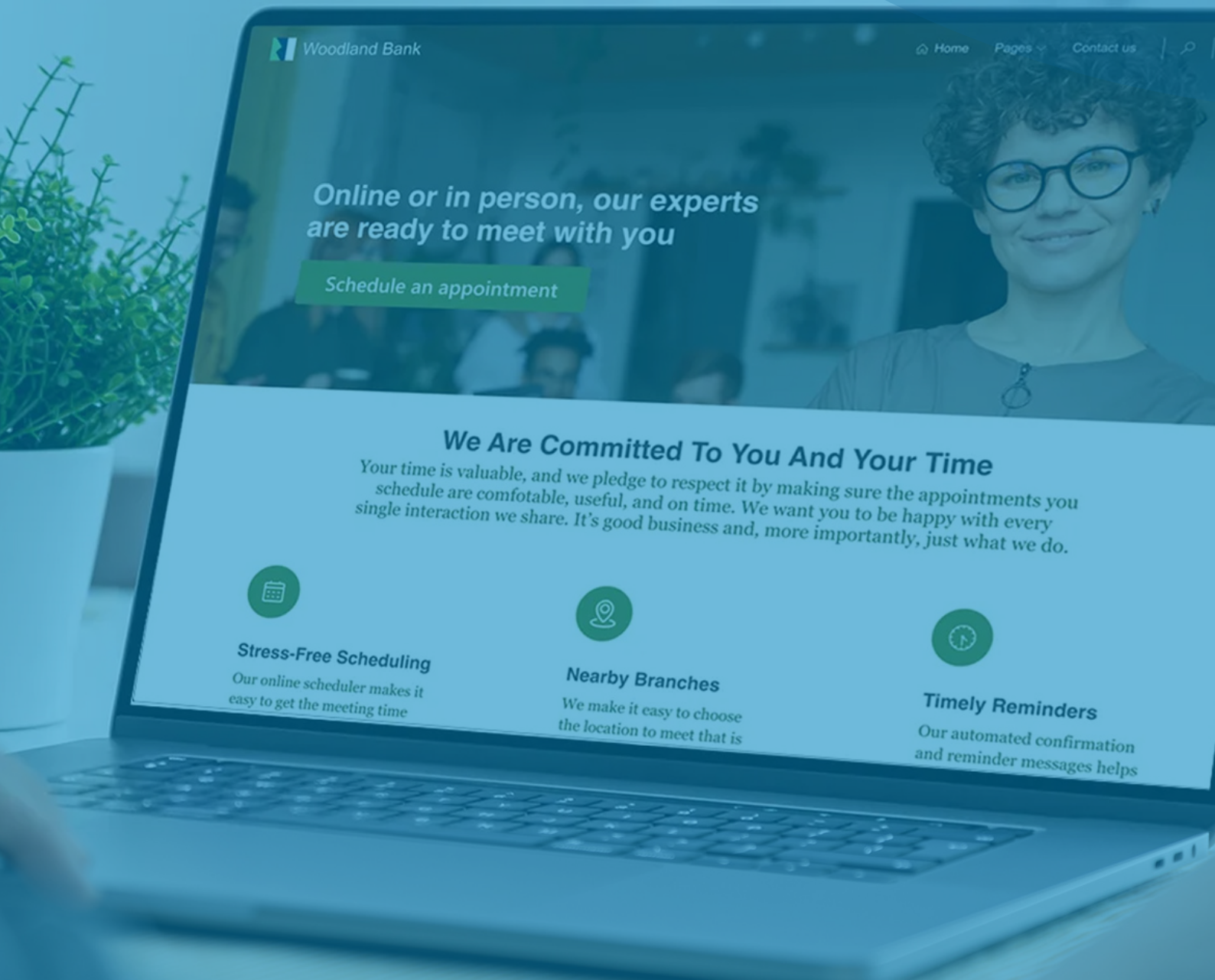
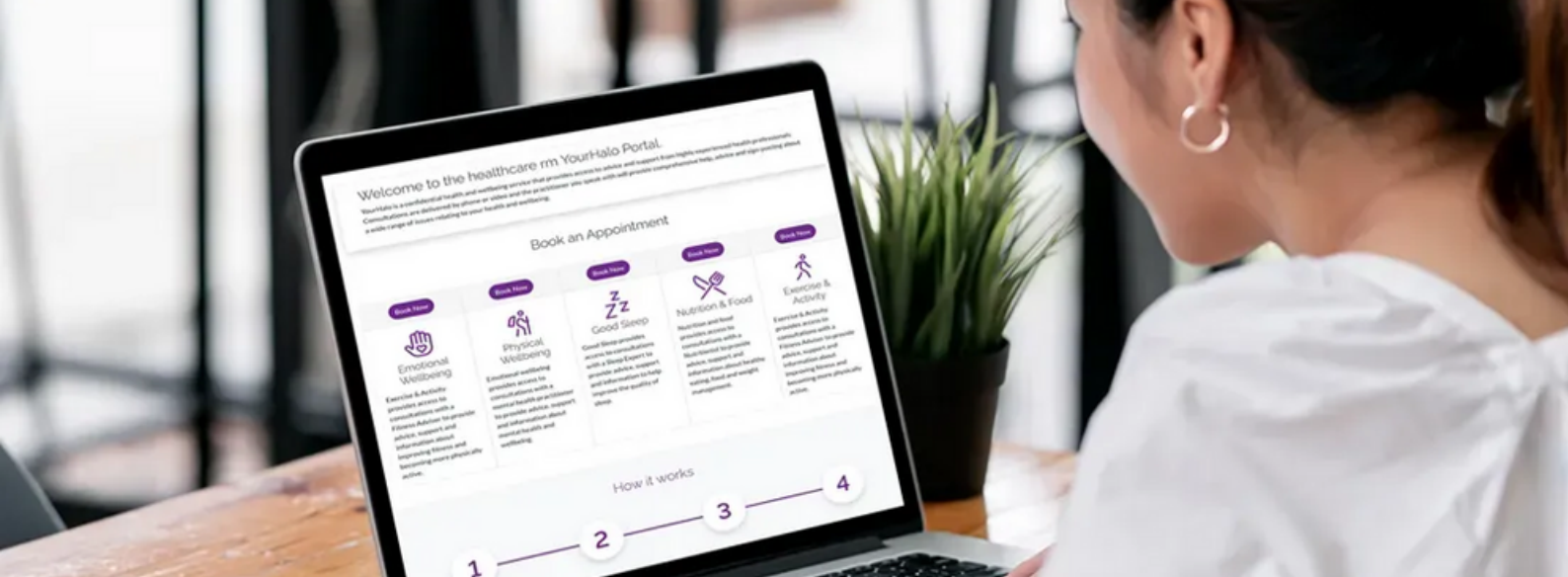




WEB PORTAL PLANNING: HOW TO PREVENT COSTLY ARCHITECTURE MISTAKES



The Hidden Costs of "Just Build It"

In the rush to launch a web portal, it's tempting to focus only on what you can see: the user interface, functionality, and your data.

The underlying portal technical architecture – the portal's foundation – is often treated as a technical detail to be handled later. This approach creates significant financial risk.

People often assume the cloud will absorb any load or that potential performance issues can be easily fixed after launch. This thinking leads to websites that perform poorly, resist re-engineering efforts, and cost significantly more to scale. The result? User frustration, missed business goals, and unbudgeted costs for emergency rework.

As specialists in designing custom web portals, we consistently witness how well-planned architecture drives long-term success.

We created this guide for project leaders to clarify your portal's technical architecture and highlight key questions to consider during the vital scoping phase that will ensure the best outcomes.

Don't worry if you can't answer all of these questions now.

Our scoping consultancy will thoroughly discuss each topic, so we can propose a portal solution that caters to your immediate and long-term needs.

We advocate a minimum viable product (MVP) framework for portal projects as it minimises upfront costs, creates early value and delivers improvements through phased releases.

Detailed scoping is an essential first step. By addressing these topics now, you can steer your project away from common pitfalls to deploy a successful, future-proof portal.

Scalability and Performance Planning

Portal performance directly impacts adoption and business outcomes. A portal that goes down or runs slowly during peak times represents failure. Early planning for growth is crucial.

User Load and Traffic Patterns

Before developers write any code, you should forecast expected user activity.

What to Discuss: How many users do you expect on day one, after year one, and in year three? Will people log in at 9 AM on a Monday, or will usage be more evenly spread each day/week? Can you identify predictable triggers that could cause portal traffic to increase 10x or even 100x the normal rate? For example, these spikes might be driven by time-critical event registrations or application deadlines.

Why it Matters: Designing a system for 10 simultaneous users fundamentally differs from handling 1,000 or more. Without planning for peak usage, it's similar to building a road without knowing whether traffic will serve a quiet town or a commuter corridor. This leads to wasteful capacities or major under-provisioning.

Scaling Strategy

This determines how easily you can add capacity to your website.

What to Discuss: Two primary methods exist: vertical scaling (making a single server more powerful, like upgrading a PC with more RAM) and horizontal scaling (adding more servers to share the load). Modern cloud applications are designed for horizontal scaling.

Why it Matters: A horizontal strategy proves more resilient and cost-effective for web portals. This approach provides flexibility, allowing you to automatically add servers during traffic spikes and remove them when normal levels return, so you only pay for the capacity you need. However, the architecture to support this should be designed from the start to avoid costly re-engineering.

Performance Requirements

To define successful outcomes, you should set specific and measurable performance targets for your portal.

What to Discuss: Replace vague goals like "the website should be fast" with concrete performance goals. For example: "The login page must load in under 1 second" or "users should be able to complete a defined journey in under 60 seconds with fewer than 5 clicks."

Why it Matters: This transforms a subjective performance wish into a measurable engineering target. It forces early architectural decisions to focus on the user experience.

Authentication and Security Architecture

How you manage portal user identity is a foundational decision with architectural implications. Changing it later will be complex and expensive, so these processes should be detailed and agreed upon in the requirements scoping stage.

Authentication Strategy

This determines how users prove who they are.

What to Discuss: Will users have a local username and password? For public portals, will you allow social logins (e.g., "Login with Google")? How will you enforce Multi-Factor Authentication (MFA)? For internal users, will individuals use their existing company credentials via Single Sign-On (SSO) for a seamless experience?

Why it Matters: These choices dictate how your portal integrates with corporate identity systems (like Microsoft Entra ID). SSO is essential for enterprise applications to reduce user friction, but requires careful planning.

Session Management

This determines how your portal remembers an authenticated user as they navigate between pages.

What to Discuss: How will you maintain sessions where multiple servers share the workload?

Why it Matters: A common architectural mistake involves storing session information on a single server, which prevents effective horizontal scaling. We recommend planning architecture for a centralised session store so that any server can handle any user's request anytime.

Data Protection and Security Framework

Your organisation's data protection policies and regulatory obligations will determine fundamental architectural choices around data storage, user access controls, and audit capabilities.

What to Discuss: How does your existing data processing policy address regulatory requirements like GDPR, and what specific data handling procedures must your portal support? What types of sensitive data will the portal process, and where can this data be stored geographically?

Do you need audit trails showing who accessed what data and when? What are your data retention requirements?

How will you build user trust through transparent data collection – will you use "just in time" notifications that explain why information is needed at the point of collection?

How should the portal integrate with your security infrastructure, such as firewalls, intrusion detection systems, or data loss prevention tools?

Why it Matters: These requirements drive critical architectural decisions that cannot easily be changed later.

Technology Stack and Integration

The tools and platforms you choose will enable or constrain your portal's future. These choices should align with your long-term goals, not just immediate familiarity.

Cloud Strategy

The cloud is not a magic bullet – its powerful tools require a clear strategy.

What to Discuss: Think beyond simply, "we'll host it on Azure." What specific services are needed? During the scoping process, we will discuss cloud services for your architecture that can automatically scale and reduce costs. Our discussions will also cover how cloud resources can be efficiently managed to minimise spending.

Why it Matters: A clear cloud strategy ensures you use the right tool for the right job, optimising performance and cost. An unplanned approach can lead to escalating monthly bills for portal usage.

Integration Requirements

Portals rarely live in isolation, so consider what data should be connected.

What to Discuss: How will the portal interface with your other systems, like Dynamics 365 CRM, ERPs, or third-party services? Will you exchange data in real-time or via nightly batch processes? What data will be accessible to each portal user role, and what actions should trigger when people use your portal to complete specific actions and processes?

Why it Matters: Poorly designed portal integrations are a primary source of performance bottlenecks and data integrity issues. You need a clear process for how data should flow between your systems and how these workflows should be managed.

Framework & Database Choice

Your portal can utilise existing data sources, like Dynamics 365 / Dataverse, or use a separate database. This choice affects performance, maintenance complexity, and integration costs.

What to Discuss: Will your portal primarily display and interact with data from existing systems like Dynamics 365, or does it need its own database for portal-specific data? If using your existing systems, what are the performance implications of real-time queries versus creating a synchronised copy of the data? What happens if your existing system is offline? What provisions should ensure the portal continues functioning with cached data?

Why it Matters: Using existing data sources like Microsoft Dataverse will reduce development time and ensure data consistency. Creating a separate database may give you more control, but it introduces data sync challenges and will increase maintenance overhead. This architectural decision affects long-term costs and system reliability, which may restrict your ability to personalise user experiences.

Web Caching and Data Management

Effective data management is essential for a high-performance portal. Fetching everything from the central database for each request causes slow page speeds and high server loads, leading to low engagement and scalability problems.

Caching Strategy

This represents one of the most effective ways to boost performance.

What to Discuss: Caching involves storing frequently accessed data temporarily in a faster location. Where can you implement this? This could be on a user's browser, on a Content Delivery Network (CDN) for global users, at the application level, or at the database level.

Why it Matters: A multi-layered caching approach can dramatically reduce load times, decrease the burden on your database, and lower infrastructure costs. It's a core element of any scalable web application.

Database Optimisation

The database serves as the heart of a web portal, but can also be a potential bottleneck.

What to Discuss: How will you maintain web performance as data volumes grow? The solution could involve query optimisation, indexing, and splitting data across multiple storage systems for better performance.

Why it Matters: These solutions shouldn't be post-launch fixes. Decisions should be made about how data is structured and partitioned early to avoid a slow, expensive, and high-risk database reorganisation project down the road.

File Upload Strategy

Your portal's file upload capabilities will directly impact user experience, storage costs, and security, requiring careful balance between functionality and system performance.

What to Discuss: What file types will users be able to upload through web forms? What maximum file sizes will you allow, and how will you balance user needs with the impact on bandwidth and cloud storage costs? Also, what timeout settings should apply to file uploads to prevent users from experiencing failures with larger files over slower connections?

How will you handle file scanning for malware and inappropriate content?

What storage duration policies will you implement – do uploaded files need to be retained indefinitely, or can they be archived or deleted after a certain period?

Why it Matters: Poor file upload policies increase security vulnerabilities, user frustration, and generate unexpected costs. These parameters are difficult to change after launch without disrupting existing users and content.

Localisation and Regional Considerations

Your portal's approach to multiple languages and regional requirements may vary depending on global scalability needs.

Multi-Language Architecture

Determining if your portal should provide multi-lingual support will impact technical and content decisions.

What to Discuss: Will your portal serve multiple languages from launch, or will this be needed later? How will you structure URLs for different languages? Where will you host the portal for optimal performance across regions? Do individual regions have specific compliance requirements, like GDPR, that will impact data storage and processing?

Why it Matters: Language and regional architecture decisions made early determine how easily you can expand into new markets without rebuilding core systems. The hosting location impacts page load speeds and may be a legal requirement for specific regions. Also, text expansion in different languages may guide user interface design and database sizing.

Monitoring and Maintenance Architecture

Your portal architecture should be sufficiently flexible and scalable to accommodate post-launch changes and ensure long-term health, easy maintenance, and rapid troubleshooting.

Observability Planning

Effective monitoring provides the visibility needed for rapid problem resolution.

What to Discuss: How will you know if your web portal is slow or has an error before people complain? The architecture must include robust logging, monitoring, and alerting.

Why it Matters: Effective monitoring can mean the difference between a 5-minute portal fix or a 5-day investigation. Observability should provide proactive insights to diagnose and resolve issues quickly, minimising downtime and protecting the user experience.

Deployment and Backups

Your portal must be easy to update and resilient to failure.

What to Discuss: How will you deploy new features without causing downtime? What is the disaster recovery plan if a server fails? How do you take data backups and, more importantly, test them?

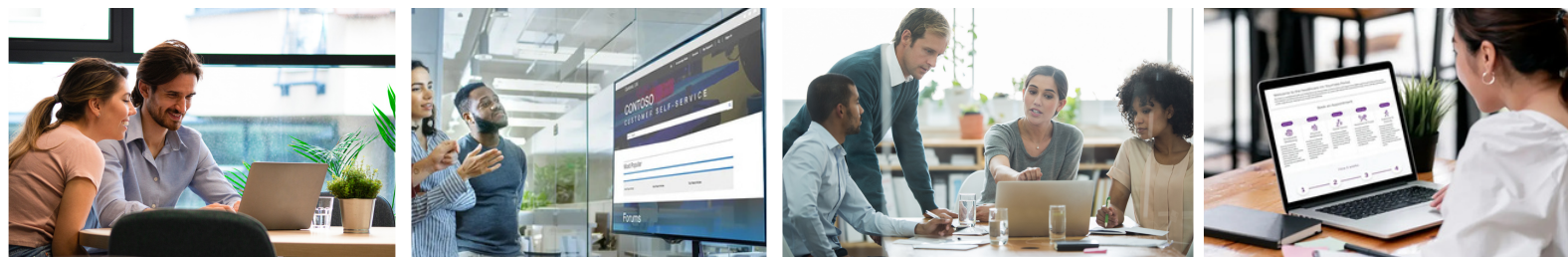
Why it Matters: These operational considerations are also dependent on architectural decisions. A portal that is difficult to update will stagnate, and a system without a reliable recovery plan represents a significant business risk.

Portal Planning Mistakes to Avoid

Engaging in portal architectural discussions early will help you avoid common obstacles.

Watch out for these red flags in your project scoping:

- **Assuming the cloud will magically solve performance issues** without a robust architecture designed to use its scaling features effectively.
- **Over-engineering with complex or trendy technologies** when a simpler solution will suffice for your requirements.
- **Under-planning for authentication**, especially in enterprise environments involving complex user roles and permission levels.
- **Neglecting caching** as an "optimisation for later." It's a core architectural component that should be planned from the start.
- **Choosing technology based on familiarity** rather than your unique needs, locking your portal into a potentially unsuitable foundation.



Your Path Forward

Technical architecture isn't an abstract IT concern – it's the foundation that determines whether your portal will deliver value and scale successfully.

Our MVP approach will get your portal live quickly and generate an early return on investment, but this doesn't mean cutting corners on architecture planning.

The decisions made during scoping will determine whether your portal can evolve seamlessly or whether you risk expensive rebuilds when subsequent changes are needed.

Effective architectural planning enables:

- Faster deployment delivery with a foundation built for growth.
- Reduced long-term costs by avoiding expensive emergency fixes and unnecessary usage.
- Enhanced user satisfaction through consistent performance from launch and phased improvements.
- Improved business agility with a data-driven portal ready for future requirements.

Successful portal projects combine rapid deployment with strong architectural frameworks – and that starts with comprehensive scoping.

Use this guide to ensure these critical discussions happen during scoping, not after launch, when architecture changes are more complex and the costs are exponentially higher.

The Portal Company specialises in helping organisations navigate these decisions to build custom portals on the Microsoft platform.

By engaging with our experts early in the process, you can avoid the frequent issues and quickly launch your portal for long-term success.

Don't let poor planning derail your project. Speak to The Portal Company today to discuss your portal requirements.

