**Пълен списък с ресурси организирани по точки**

1. **HTT****P и HTTPS**

<https://www.w3schools.in/types-of-network-protocols-and-their-uses>

<https://developer.mozilla.org/en-US/docs/Web/HTTP>

<https://www.ssl.com/faqs/what-is-https/>

<https://www.techtarget.com/whatis/definition/HTTP-Hypertext-Transfer-Protocol>

<https://www.techtarget.com/searchsoftwarequality/definition/HTTPS>

<https://www.freecodecamp.org/news/what-is-https-http-vs-https-meaning-and-how-it-works/>

<https://www.keyfactor.com/blog/http-vs-https-whats-the-difference/>

1. **Клиент сървър комуникация**

<https://www.geeksforgeeks.org/client-server-model/>

<https://www.ibm.com/docs/en/cics-ts/6.1?topic=programs-clientserver-model>

<https://en.wikipedia.org/wiki/Client_(computing)>

<https://en.wikipedia.org/wiki/Server_(computing)>

<https://www.ibm.com/docs/en/zos/2.1.0?topic=internets-typical-client-server-program-flow-chart>

<https://en.wikipedia.org/wiki/Client%E2%80%93server_model>

<https://www.quora.com/How-does-client-server-communication-work-in-web-applications>

<https://kmk.fmi-plovdiv.org/InternetTechnology.pdf>

1. **HTTP requests**

<https://learn.fmi.uni-sofia.bg/pluginfile.php/392713/mod_resource/content/2/Web%20%D1%83%D1%81%D0%BB%D1%83%D0%B3%D0%B8.%20GET%20%D0%B8%20POST%20%D0%B7%D0%B0%D1%8F%D0%B2%D0%BA%D0%B8..pdf>

<https://medium.com/@9cv9official/what-are-get-post-put-patch-delete-a-walkthrough-with-javascripts-fetch-api-17be31755d28>

<https://stackoverflow.com/questions/27030649/explain-and-example-about-get-delete-post-put-options-patch-h>

<https://dev.to/qbentil/http-methods-get-post-put-patch-delete-1fhi>

<https://www.w3schools.com/tags/ref_httpmethods.asp>

1. **HTTP Response**

<https://www.ibm.com/docs/en/cics-ts/5.3?topic=protocol-http-responses>

1. **HTML и основни тагове**

<https://www.w3schools.com/tags/ref_byfunc.asp>

1. **CSS, селектори и основни правила в CSS**

<https://developer.mozilla.org/en-US/docs/Web/CSS>

<https://www.w3schools.com/css/css_intro.asp>

<https://www.w3.org/TR/2018/REC-selectors-3-20181106/>

<https://en.wikipedia.org/wiki/CSS>

<https://www.w3schools.com/tags/ref_byfunc.asp>

1. **Семантични страници**

<https://bg.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D1%87%D0%B5%D0%BD_HTML>

<https://savovdesign.com/html5-semantichni-elementi-i-polzata-im-za-seo>

<https://webfe.wordpress.com/html/%D1%81%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D1%87%D0%B5%D0%BD-html/>

<https://www.semrush.com/blog/semantic-html5-guide/>

<https://medium.com/before-semicolon/10-html-semantic-tags-and-when-to-use-them-5ae7d7d0b0f2>

<https://primeinspire.com/blog/new-semantic-elements-in-html5-with-examples>

<https://stackoverflow.com/questions/9563845/semantic-use-of-nav-in-html5-with-search-form-element>

1. **Адаптивно (responsive) оформление на страници**

<https://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/#:~:text=Responsive%20Web%20design%20is%20the,use%20of%20CSS%20media%20queries>.

<https://www.w3schools.com/html/html_responsive.asp>

<https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design>

<https://en.wikipedia.org/wiki/Responsive_web_design>

1. **JavaScript**

<https://en.wikipedia.org/wiki/JavaScript>

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

1. **Bootstrap**

<https://getbootstrap.com/docs/3.4/css/>

<https://getbootstrap.com/2.3.2/base-css.html>

<https://getbootstrap.com/docs/4.0/content/typography/>

1. **Сървърни езици**

<https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Introduction>

<https://www.qulix.com/about/blog/the-best-server-side-language/>

<https://www.ibm.com/docs/en/i/7.3?topic=programming-server-side-scripting-languages>

<https://www.geeksforgeeks.org/server-side-client-side-programming/>

1. **База данни и СУРБД**

<https://www.oracle.com/database/what-is-database/#:~:text=Is%20a%20Database%3F-,Database%20defined,database%20management%20system%20(DBMS)>.

<https://bg.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D0%B8>

<https://bg.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%B7%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BD%D0%B0_%D0%B1%D0%B0%D0%B7%D0%B8_%D0%BE%D1%82_%D0%B4%D0%B0%D0%BD%D0%BD%D0%B8>

<https://obuch.info/bazi-danni-v2.html>

<http://tru.uni-sz.bg/virtru/Relations.htm>

<https://codegym.cc/bg/quests/lectures/bg.questhibernate.level17.lecture02>

<https://help.claris.com/archive/help/18/fmp/en/index.html#page/FMP_Help/one-to-one-relationships.html>

<https://blog.devart.com/types-of-relationships-in-sql-server-database.html#:~:text=So%2C%20what%20is%20one-to,one%20record%20in%20table%201>

<https://www.upgrad.com/blog/types-of-keys-in-dbms/>

<https://www.upgrad.com/blog/code-first-approach-in-mvc/>

<https://www.tutorialspoint.com/entity_framework/entity_database_first_approach.htm>

1. **UML и E/R диаграми**

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/>

<https://en.wikipedia.org/wiki/Unified_Modeling_Language>

<https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/>

<https://www.lucidchart.com/pages/er-diagrams>

<https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model>

1. **Многослойна архитектура**

<https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html>

<https://stackoverflow.com/questions/8336449/architecture-principles-for-how-to-create-multiple-layers>

<https://niteco.com/articles/5-principles-for-software-architecture/>

<https://www.sciencedirect.com/topics/computer-science/tier-architecture>

1. **Трислойни модели**

<https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/architectural-principles>

<https://medium.com/@syantien/clean-architecture-5574cd731e8d>

<https://www.codemag.com/article/0705071/Layered-Architecture-Dependency-Injection-and-Dependency-Inversion>

<https://medium.com/geekculture/what-is-a-layered-architecture-in-software-design-22152fc06822>

<https://cw.fel.cvut.cz/old/_media/courses/b6b33ear/lecture-04-architectures-h.pdf>

<https://www.geeksforgeeks.org/mvc-design-pattern/>

<https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/>

<https://towardsdatascience.com/everything-you-need-to-know-about-mvc-architecture-3c827930b4c1>

<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>

<https://dev.to/sardarmudassaralikhan/clean-architecture-used-in-software-development-5gpc>

<https://www.toptal.com/software/single-responsibility-principle>

<https://hackernoon.com/model-view-controller-architecture-pattern-usage-advantages-examples>

<https://crimsonpublishers.com/prsp/fulltext/PRSP.000505.php>

<https://medium.com/@charithavenkataraju/model-view-controller-mvc-ba6c07dff565>

<https://stackoverflow.com/questions/16634501/in-proper-mvc-does-everything-have-to-be-a-model-view-or-controller>

<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel>

<https://www.ramotion.com/blog/what-is-mvvm/>

<https://www.netguru.com/blog/mvvm-architecture>

<https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm>

<https://builtin.com/software-engineering-perspectives/mvvm-architecture>

<https://dev.to/mochafreddo/mastering-mvvm-a-comprehensive-guide-to-the-model-view-viewmodel-architecture-221g>

<https://www.geeksforgeeks.org/mvvm-model-view-viewmodel-architecture-pattern-in-android/>

<https://medium.com/@hussainabbas365/mvvm-with-clean-architecture-in-react-native-a-detailed-guide-3ff387944692>

<https://copyprogramming.com/howto/mvvm-ddd-and-wpf-layered-application-project-structure-guidance>

<https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html>

<https://www.brcline.com/blog/introduction-to-the-layered-architecture-n-tier-architecture>

<https://www.linkedin.com/pulse/decoding-few-technical-jargons-recruiters-frontend-backend-pandey>

<https://www.ibm.com/topics/three-tier-architecture>

<https://www.ibm.com/topics/three-tier-architecture>

<https://www.brcline.com/blog/introduction-to-the-layered-architecture-n-tier-architecture>

<https://www.geeksforgeeks.org/mvc-framework-introduction/>

<https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm>

<https://www.techtarget.com/whatis/definition/Model-View-ViewModel#:~:text=MVVM%20is%20also%20known%20as,of%20code%20simpler%20and%20faster>

<https://learn.microsoft.com/en-us/xamarin/xamarin-forms/enterprise-application-patterns/mvvm>

<https://www.tutorialspoint.com/mvvm/index.htm>

<https://www.codeproject.com/Articles/100175/Model-View-ViewModel-MVVM-Explained>

<https://www.guru99.com/mvc-vs-mvvm.html>

<https://in.indeed.com/career-advice/career-development/difference-between-mvvm-and-mvc>

<https://startup-house.com/blog/mvvm-vs-mvc-comparison>

1. **Обектно-релационно свързване (ORM frameworks)**

<https://www.freecodecamp.org/news/what-is-an-orm-the-meaning-of-object-relational-mapping-database-tools/>

<https://bg.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D0%B8>

<https://obuch.info/lekcii-po-bazi-danni.html?page=2>

<https://www.hwlibre.com/bg/orm-object-relational-mapping/>

1. **Упълномощаване (authentication) и удостоверяване (authorization)**

<https://indico.cern.ch/event/63130/contributions/1225280/attachments/1007198/1432928/BG_AA-bg-1.pdf>

<https://www.mtc.government.bg/bg/node/4751>

<https://bg.surveillancepackages.com/difference-between-authentication-and-authorization-4b26>

<https://www.onelogin.com/learn/authentication-vs-authorization#:~:text=Authentication%20and%20authorization%20are%20two,authorization%20determines%20their%20access%20rights>.