

1. Web Proqramlaşdırma Giriş və İnternet Əsasları

- **İnternet və Şəbəkə Əsasları**
 - İnternetin işləmə prinsipi, şəbəkə arxitekturası
 - **OSI və TCP/IP modelləri**, əsas protokollar (HTTP, FTP, TCP, UDP)
 - **TCP və UDP Arasındakı Fərqlər**
 - **TCP** və **UDP** protokollarının müqayisəsi
 - Hansi hallarda hansı protokoldan istifadə edilir
 - Hosting, domenlər, server və müştəri arxitekturası
- **Veb Texnologiyaları və HTTP**
 - HTTP və HTTPS protokolları, SSL/TLS sertifikatları
 - **DNS**, URL strukturları, URI və URN fərqləri
 - **HTTP metodları**: GET, POST, PUT, DELETE, PATCH, OPTIONS, HEAD
 - **HttpRequest**, **HttpResponse** strukturu, başlıqlar və status kodları
 - **CORS (Cross-Origin Resource Sharing)**
 - CORS prinsipləri və **@CrossOrigin** annotasiyası
 - **CorsConfiguration** və **CorsFilter** ilə tənzimləmə
- **Server və Servletlər**
 - Server, Handler, Mapping, **Servlet** və Filter-lər
 - Java EE ilə ilk veb tətbiq yaradılması

2. Java EE və Spring Framework Əsasları

- **Java EE Əsasları**
 - Java Servlets, JSP, JSF və onların istifadəsi
 - **EJB (Enterprise Java Beans)**, JPA (Java Persistence API)
- **Spring Framework Əsasları**
 - **Inversion of Control (IoC)** və **Dependency Injection (DI)**
 - ApplicationContext və Bean Lifecycle
 - **Spring** və **Spring Boot** arasında fərqlər
 - Java EE-dən Spring Boot-a keçid yolları

3. Spring Boot Əsasları və Layihə Quruluşu

- **Spring Boot-un Əsas Konsepsiyaları**
 - **Auto-Configuration**, Starter-lər və onların rolu
 - **Conventional Project Structure**
 - Maven və Gradle ilə layihə idarəetməsi
 - **Infra Package** və onun strukturu
- **Beans və Bean Konfigurasiyası**
 - Bean yaradılması, **scope-lar** (Singleton, Prototype)
 - **@Component**, **@Service**, **@Repository**, **@Controller** annotasiyaları
 - **Component Scan** və Package Structure
 - **Open-in-View** konsepti və istifadəsi

4. Spring Boot Annotasiyaları və Konfigurasiya

- **Əsas Annotasiyalar**

- @SpringBootApplication, @EnableAutoConfiguration, @SpringBootConfiguration
- @ComponentScan, @Configuration, @Bean
- **Controller və Endpoint Annotasiyaları**
 - @RestController, @Controller, @RequestMapping, @GetMapping, @PostMapping, @PutMapping, @DeleteMapping
 - @ResponseBody, @ResponseStatus
 - @PathVariable, @RequestParam, @RequestBody, @RequestHeader, @CookieValue
- **Konfigurasiya və Properties İdarəetməsi**
 - application.properties və application.yaml faylları
 - @Value, @ConfigurationProperties
 - **Profilər** və mühitlər üzrə konfigurasiya
 - **prefer-ip-address** parametrinin araşdırılması

5. Project Lombok və Spring Validation

- **Project Lombok ilə Kodun Sadələşdirilməsi**
 - @Getter, @Setter, @ToString, @EqualsAndHashCode, @Data
 - @NoArgsConstructor, @AllArgsConstructor, @Builder
- **Məlumatların Validasiyası**
 - **Bean Validation API**
 - @Valid, @Validated
 - Validasiya annotasiyaları: @NotNull, @NotBlank, @NotEmpty, @Min, @Max, @Size, @Email, @Pattern
 - **Custom validator**-lərin yaradılması

6. Global Exception Handling və Loglama

- **Səhv İdarəetməsi**
 - @ControllerAdvice, @RestControllerAdvice
 - @ExceptionHandler, **Custom Exception**-lərin yaradılması
 - Hata mesajlarının və status kodlarının özəlləşdirilməsi
- **Loglama**
 - **Loglama framework**-ləri: Log4j2, SLF4J
 - Log səviyyələri: ERROR, WARN, INFO, DEBUG, TRACE
 - Loglama konfigurasiyası və loqların idarə olunması

7. Fayllarla İşləmə və Statik Resurslar

- **Fayl Yükləmə və Yüklənməsi**
 - **Multipart faylların** qəbul edilməsi və işlənməsi
 - Faylların serverdən yüklənməsi
 - Fayl saxlama strategiyaları (lokal, bulud əsaslı)
- **Statik Resursların İdarə Olunması**
 - CSS, JS və şəkil fayllarının idarə olunması

8. Spring Web və Front-end İntegrasiyası

- **Thymeleaf ilə Şablonların Hazırlanması**
 - **Thymeleaf sintaksisi** və ifadələri
 - Formaların yaradılması və göndərilməsi
 - **Fragment**-lər və şablonların təkrar istifadəsi
- **Internationalization və Lokalizasiya**
 - Mesaj resursları və dillərarası dəstək
 - **Locale** seçimi və idarə edilməsi

9. Data Access Layer: JDBC, JPA və Hibernate

- **JDBC Template ilə İşləmə**
 - **JdbcTemplate** və **NamedParameterJdbcTemplate**
 - **RowMapper**-lər və **ResultSetExtractor**-lər
- **ORM və Hibernate Əsasları**
 - ORM konsepti, **Hibernate**-in iş prinsipi
 - **Entity**-lər, **Session**, **Transaction**
- **Spring Data JPA ilə İntegrasiya**
 - **JpaRepository**, **CrudRepository** istifadəsi
 - Custom query-lər, **@Query** annotasiyası
 - **Query Methods** və **Method Names**
- **H2 Database ilə Test Mühiti**
 - **H2 in-memory database** istifadəsi
 - Testlər üçün data init skriptlərinin yaradılması

10. Məlumat Bazasındaki Əlaqələr və Mapping

- **Entity Əlaqələri**
 - **@OneToOne**, **@OneToMany**, **@ManyToOne**, **@ManyToMany**
 - **Cascade Type**-lər, **Fetch Type**-lər (Lazy, Eager)
 - **Orphan Removal**, **Join Column**-lər
- **Advanced Mapping**
 - **Inheritance Mapping** (**@Inheritance**)
 - **Embeddable** və **Embedded** obyektlər
 - **Native Query**-lər və **Stored Procedure**-lər

11. Verilənlər Bazası Dəyişikliklərinin İdarə Olunması

- **Liquibase ilə Migration**
 - Liquibase konsepti, **ChangeSet**-lər və **ChangeLog**-lər
 - SQL, XML, YAML formatlarında dəyişikliklər
 - **Rollback** əməliyyatları və **Version Control**
- **Semantic Versioning (semver)**
 - Versiyalaşdırma strategiyaları və ən yaxşı təcrübələr

12. Təhlükəsizlik: Authentication və Authorization

- **Spring Security Əsasları**
 - **Authentication** və **Authorization** konseptləri
 - **SecurityConfig** və **WebSecurityConfigurerAdapter**

- **UserDetailsService** və Custom User Details
- **Şifrə İdarəetməsi**
 - **Password Encoding**, BCryptPasswordEncoder
 - Şifrə reset və dəyişdirmə funksiyaları
- **Role-based Access Control**
 - **Authorities** və **Roles**
 - @PreAuthorize, @PostAuthorize, @Secured annotasiyaları
- **CORS və CSRF Qorunması**
 - CORS prinsipləri və @CrossOrigin annotasiyası
 - **CorsConfiguration** və **CorsFilter** ilə tənzimləmə
 - **CSRF (Cross-Site Request Forgery)** hücumları və qorunma yolları
- **XSS və SQL Injection**
 - XSS və SQL Injection hücumları
 - **Input validasiyası** və parametrlı sorğular

13. JWT və OAuth2 ilə Təhlükəsizlik

- **JWT (JSON Web Token) İnteqrasiyası**
 - JWT konsepti və strukturu
 - **Access Token** və **Refresh Token** yaradılması və idarə olunması
 - Token expiration, yenilənmə və **blacklisting**
- **OAuth2 və OpenID Connect**
 - OAuth2 axışı və **grant** tipləri
 - **Authorization Server** və **Resource Server**
 - OpenID Connect ilə identifikasiya
 - Spring Security OAuth2 ilə implementasiya
- **Encryption (Symmetric / Asymmetric)**
 - Simmetrik və assimetrik şifrələmə
 - **SSL/TLS** və sertifikatların istifadəsi

14. API Dizaynı və İdempotensiya

- **RESTful API Dizayn Prinsipləri**
 - Resource-lar, URL strukturları, HTTP metodları
 - **HATEOAS** konsepti
 - **API First Design** yanaşması
- **İdempotensiya və Təhlükəsiz HTTP Metodları**
 - İdempotent əməliyyatlar və onların tətbiqi
 - **Retry** mexanizmləri və **idempotency keys**
 - Concurrency problemləri və **optimistic locking**
- **OpenAPI və Swagger ilə Sənədləşdirmə**
 - API-lərin avtomatik sənədləşdirilməsi
 - **SpringDoc** və **springfox-swagger** ilə inteqrasiya

15. Microservice Arxitekturası və Spring Cloud

- **Software Architecture Overview**
 - **Monolit, n-Layered Architecture, SOA, Microservices**
 - **CAP Teoremi** və **ACID** prinsipləri

- **Coupling** (tight coupling) və **Cohesion**
- **Decomposition** strategiyaları
- **Fast Initial Development** və çevik yanaşma
- **Microservice Arxitekturası**
 - Mikroservislərin üstünlükləri və çətinlikləri
 - **Service Discovery** və **Eureka Server**
 - **Spring Consul** və **Zookeeper**
- **API Gateway və BFF**
 - **Netflix Zuul Gateway**
 - **Spring Cloud Gateway**
 - **Kong Gateway**
 - **API Gateway** və **BFF (Backend for Frontend)** arasındakı fərqlər
- **Spring Cloud Komponentləri**
 - **Configuration Management**: Spring Cloud Config Server
 - **Circuit Breaker** (Hystrix, Resilience4j)
 - **Spring Cloud** və **Spring Boot** versiya uyğunluğu
- **Distributed Tracing və Monitoring**
 - **Spring Cloud Sleuth, Zipkin**
 - Metrics və Health Checks
- **Deployment Strategiyaları**
 - **Blue/Green** və **Canary Deployment**

16. Asinxron Proqramlaşdırma və Mesajlaşma

- **Asinxron İşlər və @Async**
 - **@EnableAsync** və **@Async** annotasiyaları
 - **Thread Pool**-lərin konfigurasiyası
- **Mesajlaşma Sistemləri**
 - **JMS** (Java Messaging Service)
 - **RabbitMQ** və **Kafka** ilə integrasiya
 - **Spring AMQP** və **Spring Kafka**

17. Caching və Performans Artırma

- **Cache Strategiyaları**
 - Cache konsepti və caching-in üstünlükləri
 - **@Cacheable**, **@CacheEvict**, **@CachePut** annotasiyaları
- **Redis və EhCache İntegrasiyası**
 - **Redis** nədir və necə işləyir
 - **Spring Data Redis** ilə integrasiya
 - Cache konfigurasiyası və idarə olunması
- **Performansın Optimallaşdırılması**
 - Profiling, **Snyk** ilə təhlükəsizlik skanərləri
 - **Fig Pattern** və dizayn nümunələri

18. Testing və Keyfiyyət Təminatı

- **Unit və Integration Testing**
 - **JUnit 5**, **Mockito** ilə testlərin yazılması

- **MockMvc** ilə web layer testləri
- **Embedded Database** ilə data layer testləri
- **Continuous Integration (CI) və Continuous Deployment (CD)**
 - CI/CD-nin faydaları (insan faktorunu aradan qaldırmaq, effektivlik artırmaq)
 - **Jenkins, GitLab CI/CD, GitHub Actions**
 - **Blue/Green** və **Canary Deployment** strategiyaları
 - **Semantic Versioning (semver)** və versiyalaşdırma

19. Containerization və Deployment

- **Docker ilə Konteynerləşdirmə**
 - Docker konsepti, **Dockerfile** yaradılması
 - Docker Compose ilə multi-container tətbiqlər
- **Kubernetes Əsasları**
 - **Pod, Deployment, Service, Ingress**
 - ConfigMap, Secret-lər və onların idarə olunması
 - Kubernetes-də Spring Boot tətbiqlərinin yerləşdirilməsi
- **Infra as Code** və avtomatlaşdırma

20. Əlavə Mövzular və Praktiki Məsələlər

- **Snyk** ilə təhlükəsizlik skanerləri və zəifliklərin aşkarlanması
- **Fig Pattern** və dizayn nümunələri
- **Decomposition** və mikroservislərin bölünməsi
- **Monolit və Service Architecture**-lərin müqayisəsi
- **Prefer-ip-address**
parametrinin araşdırılması və tətbiqi