**Birat Kshitiz College**



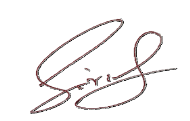
Shahi Marga, Biratnagar

Faculty of Humanities and Social Sciences

**INTERNSHIP –STUDENT’S WEEKLY LOG**

**TU Registration No.: 6-2-1074-0012-2019**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Time** | | | **Main Task** | **Activities** |
| **In** | **Out** | **Hour**  **(per day)** |
| Jan7 – Jan16 | 10 AM | 3 PM | 5 | **Introduction to Spring, Maven and Hibernate** | ●Read introductory material on Spring framework.  ●Explore dependency injection and inversion of control concepts in spring. ●Learn the fundamentals of maven, Understand maven’s project structure and Lifecycle.  ●Study the basics of hibernate ORM.  ●Explore object-relational mapping concept. |
| Jan17 – Jan24 | 10 AM | 3 PM | 5 | **Building basic Spring Boot , Web Application** | ●Setup spring boot project using spring initializer.  ● Configured dependencies (MYSQL Driver, Spring Data JPA , Spring Boot Actuator, Spring Security, Spring Boot DevTools , Spring Web).  ●Configured application properties for database connectivity.  ●Integrated Bootstrap for styling and responsive design. |
| Jan25- Feb1 | 10 AM | 3 PM | 5 | **User management module** | ●Created the necessary entity classes to represent user data i.e. user.java and role.java.  ●Defined relationships between entities using annotations like @OneToMany, @ManyToOne, etc.  ●Created service interface and implementation classes i.e. UserService.java, UserServiceImpl.java, Rolesevice.java and RoleServiceImpl.java to handle business logic related to user management.  ●Created JSP page i.e. user.java and role.java to implemented endpoints for CRUD operation and mapped them to appropriate service methods.  ●Created controller i.e.UserController.java and RoleController.java to handle HTTP requests related to user operation  ●Configured spring security to protect endpoints and enforce authentication.  ●Set-up role-based access control to manage user permissions. |
| Feb2- Feb9 | 10 AM | 3 PM | 5 | **Category and Product Module** | ●Created entity classes i.e. category.java and product.java.  ●Defined relationships between entities using annotations like @OneToMany, @ManyToOne, etc.  ●Created service interface and implementation classes i.e. CategoryService.java, CategoryServiceImpl.java, Productsevice.java and ProductServiceImpl.java.  ●Created JSP page i.e. category.java and product.java to implemented endpoints for CRUD operation and mapped them to appropriate service methods.  ●Created controller i.e.CategoryController.java and ProductController.java to handle HTTP requests and facilitate the interaction between the frontend and backend web application. |
| Feb11 – Feb18 | 10 AM | 3 PM | 5 | **Cart and Order Module** | ●Created entity classes i.e. cart.java and order.java.  ●Defined relationships between entities using annotations like @OneToMany, @ManyToOne, etc.  ●Created service interface and implementation classes i.e. CartService.java, CartServiceImpl.java, Orderevice.java and OrderServiceImpl.java.  ●Created JSP page i.e. cart.java and order.java to implemented endpoints for CRUD operation and mapped them to appropriate service methods.  ●Created controller i.e.CartController.java and OrderController.java to handle HTTP requests and facilitate the interaction between the frontend and backend web application. |
| Feb20 – Feb28 | 10AM | 3PM | 5 | **Payment System Integration** | ● Studying eSewa integration involves understanding various aspects such as its technical architecture, implementation methods, user experience etc.  ● Created PaymentController class which plays a central role in managing payment-related operations.  ● Created payForOrder method in controller manages payment processing, calculating total amounts, generating transaction UUIDs, and crafting secure signatures using HMAC-SHA256, enriching the model with attributes for rendering the "epayment" view.  ● Created service interface and implementation classes i.e. PaymentService.java and PaymentServiceImpl.java. The service layer is responsible for encapsulating the business logic related to eSewa integration. |
| Feb29-March7 | 10AM | 3PM | 5 | **Build REST API** | ● Study the basics of REST API.  ● Created an entity class to represent our data model.  ● Created a repository interface to handle database operations for the entity.  ● Created a service class to handle the business logic for our REST API.  ●Created a controller class to handle HTTP requests for our REST API. |
| Feb29-March7 | 10AM | 3PM | 5 | Sending Email via SMTP | ● Study and understand how can we send email via smtp.  ● Adding the **spring-boot-starter-mail** dependency in **pom.xml.**  **●**SettingUp **Application.properties** file with configurations required for using **Gmail SMTP** server.  ● Creating EmailDetails class that contains fields such as recipient, msgBody, subject, and attachment.  ● Creating interface EmailService  And implementing class  EmailServiceImpl of service layer  ●Creating Rest Controller EmailController which defines various API for sending email. |

**Supervisor Signature:**  **Date:**