

SMU IS621 Agile and DevSecOps
Final Project Report



Learn N Play
Team 5

Chow Hui Ling
Foo T'suen Wei
Kruti Chandrashekhar
Li Shuangshuang
Priya Haresh Lalla
Rohan Mehta

CONTENTS

Project Overview	1
Vision.....	1
Approach	1
Collaboration Tools.....	1
Member Roles	1
Agile Practices.....	1
Overview.....	1
Story Point Estimation	3
Technical Details	4
Software Architecture	4
Security Considerations.....	7
User Interface	8

Project Overview

Vision

Empowering Singaporean parents with affordable educational toys. Our project aims to establish a user-friendly website for selling quality educational toys to parents in Singapore that can help children learn and play at the same time.

Approach

Instead of building from scratch, the team chose to adapt an existing open-source project for our use case. This would allow us to test the market to gauge the demand for educational toys outside our own social network, at a relatively lower upfront cost. The first step was to look for a viable code base that we could deploy and build on. The team shortlisted [Odoo](#), [Evershop](#), [OpenCart](#), and a few others. Evershop was the first choice due to its smaller code base which made it simpler to understand and test. @Foo volunteered to deploy Evershop on his private server for the project, and the team agreed since it seemed sufficient to test our market viability. @Foo soon found Evershop to be relatively buggy within the first few days and quickly pivoted to deploy [OpenCart](#), which was the next simplest product that seemed fully featured for our use case.

Collaboration Tools

Jira, Confluence, MS Teams, WhatsApp

Member Roles

This is how we split the roles, with each team member assuming an internal role (to state the business need) and acting as part of the scrum team (to test/enhance the features):

Scrum Roles	Names	Internal Roles	Responsibilities
Product Owner	Chow Hui Ling	Operation Manager	Responsible for defining the project's vision, prioritizing the product backlog, and ensuring that the development aligns with user needs and company goals. Additionally, testing functionalities related to customer reviews, promotional activities, and logistics management to ensure a seamless shopping experience.
Scrum Master	Foo Tsuen Wei (Foo)	Software Architect	Responsible for the infrastructure setup and facilitation of scrum meeting discussions. Also fixes code raised by the team.
Developer	Li Shuangshuang	Business Manager	Responsible for defining business requirements and also serving as an admin to test functionalities such as category and product editing, as well as dashboard displays.
Developer	Priya Lalla	Operation & Maintenance Engineer	Responsible for the CI/CD setup of the product and fixing of bugs found (if any)
Developer	Kruti Chandrashekhar	UI Designer	Responsible for Name, logo design, theme colour and website user interface arrangements. Also responsible for testing the shopping experience from customers' perspectives.
Developer	Rohan	Information Security Engineer	Responsible for the overall architecture and overseeing the security aspects of the product. Also helps in fixing code issues.

Figure 1 Member Roles

Agile Practices

Overview

After the initial brainstorming and pitching of ideas, the project vision and approach above were decided through our initial **project kick-off meeting**. The **product backlog** was first drafted on Confluence collectively then transferred to Jira after discussion and agreement of the relative priorities of the stories. We wrote minimal info to start off discussions and get mutual understanding of the tasks. We organised the stories using the following epics, with lower priority features marked as *:

Epic	Remarks
Seller - Catalogue Management	includes brand, product details
Shopping Cart and Checkout Process	N/A
Seller - Order Management and fulfilment	Stock status, tracking orders, status updates
Seller - UI Customization	Display settings, logo, colours, themes
Search and Navigation	Includes search, filter, sort, product listing, description
User Authentication and Account Management	Registration, user preferences
Seller - Marketing and Promo	Includes discounts, promotion, featured items, marketing dashboard
Tech Operations	Technical documentations and Design, CI/CD, backup and recovery, SLA and response time
Payment Gateway Integration	Actual integration with payment gateway
User Reviews and Ratings	N/A
Support	Enquiry, review, exchanges, returns, feedback, FAQ
Product Sharing *	One-click share to social media
Loyalty Programme *	Loyalty points
Product Recommendations*	N/A

Figure 2 Epics Used to Organize the User Stories

Figure 3 below shows the summary of the two sprints that the team went through.

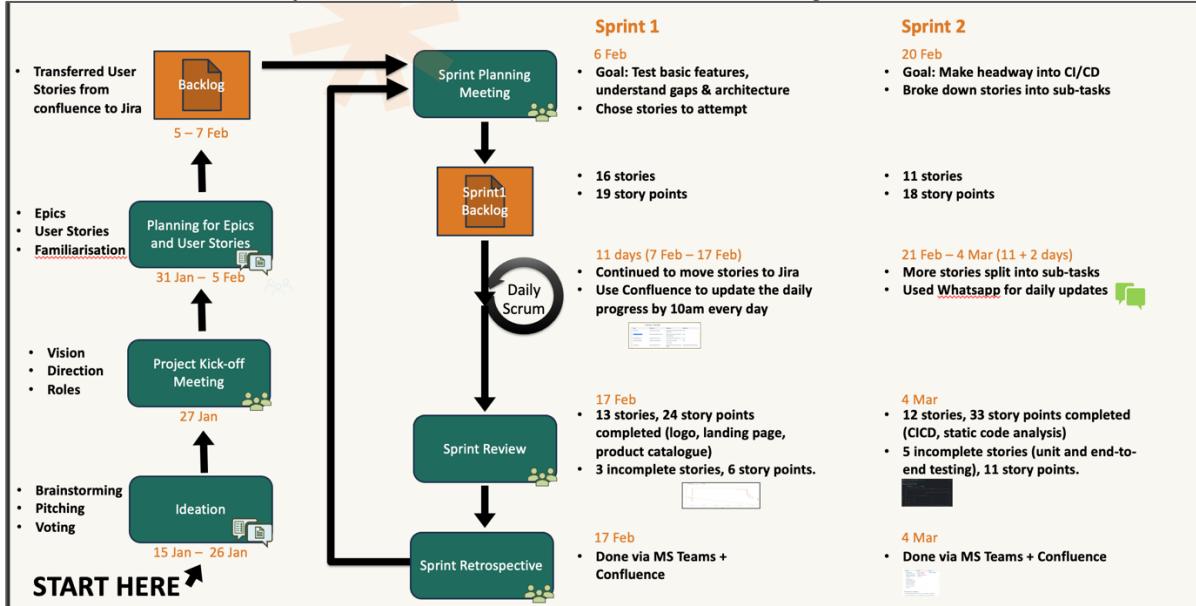


Figure 3 Overall Agile Process with Two Sprint Cycles

Goal of Sprint 1 was to test basic functionalities as well as understanding the architecture of OpenCart, so as to identify any gaps that needed to be remedied. An example of the **acceptance criteria** of a feature-testing user story is in Figure 14 in the Appendix. At the end of each test, missing features or gaps in the functionalities were to be added as additional stories that we would need to implement in subsequent sprints. The team held a **sprint planning** meeting, breaking down stories into sub-tasks but without providing hourly estimates due to task unfamiliarity. **Daily progress updates** were input into Confluence by 10am daily as we could not find common timeslot to meet daily. See Figure 15 in Appendix for example of the updates for one of the days. At the **sprint review**, the “Website logo and UI design” story was further broken down so that we could mark the first half (shop logo) as completed, and the other half (UI themes) to be pushed to backlog. (See Figure 20 and Figure 21 in Appendix for details). The Sprint Burndown Chart in Figure 16 also revealed that a lot of the stories were closed near the end of sprint, and that some estimates went up following a better understanding of some of the task complexities during the sprint. These were not too concerning as the team continued to keep each other updated on the progress via WhatsApp.

The sprint velocity was approximately 24 story points, with some stories dropped or pushed to next sprint due to complexity. The **sprint retrospective** highlighted a common consensus to shift daily updates to WhatsApp for faster

communication, due to the notification features that allowed for faster responses over Whatsapp. We also started to appreciate the potential benefits of setting up a CI/CD pipeline, and automated end-to-end testing to ease our subsequent testing efforts when we went on to implement the missing functionalities we had identified.

Hence in Sprint 2, we aimed to make headway into CI/CD. Sprint ceremonies were similar to Sprint 1, except that this time round, we used Whatsapp instead of Confluence for our daily updates. We also tried to break down the stories upfront into more subtasks so that we could better estimate the complexity for each task. At the end of the sprint, we had managed to set up an initial CI/CD pipeline and implemented some automated scanning, though additional fine-tuning had to be made. Details of the technical implementations are given in the Technical Details section.

This is the progress of the backlog by the end of the two sprints:

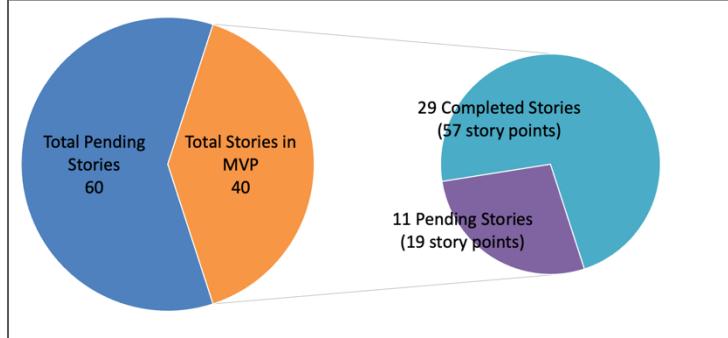


Figure 4 Overall Status of Backlog Showing Number of Stories (Total and MVP) and Story Points Remaining and Completed (for MVP)

A screenshot of a Jira Issues page titled 'Team 5'. The search bar shows the query: 'project = "TS" and issuetype != Epic and status not in (Done, "Won't Do") and fixVersion = MVP ORDER by Parent, created ASC'. The results table lists 11 issues, all under the 'MVP' fix version. The issues are categorized by parent epic: 'User Reviews and ratings' (T5-109), 'User Authentication and Account Manag' (T5-17), 'Tech Operations' (T5-119), 'Seller - Order Management and Fulfilme' (T5-100), 'Seller - Order Management and Fulfilme' (T5-99), 'Seller - Order Management and Fulfilme' (T5-98), 'Search and Navigation' (T5-105), 'Search and Navigation' (T5-103), 'Payment Gateway Integration' (T5-108), 'Helpdesk and Support' (T5-79), and 'Helpdesk and Support' (T5-78). The page shows 1-11 of 11 issues, with a single page indicator.

Figure 5 Team's current backlog screenshot (Remaining stories for) Refer to Appendix to see more stories.

Story Point Estimation

The team used Jira Planning Poker for the story point estimation at the start of each sprint planning session. Our sprint velocity was 24 and 33 story points respectively for the first and second sprint. Considering that we still had about 19 story points for our MVP, and that we expect additional new situations that we haven't encountered yet (e.g. additional security vulnerabilities encountered from security scans that we need to fix, and the fact that the team tends to underestimate the task complexity, we expect to take at least two more sprints to delivery our MVP.

Technical Details

Software Architecture

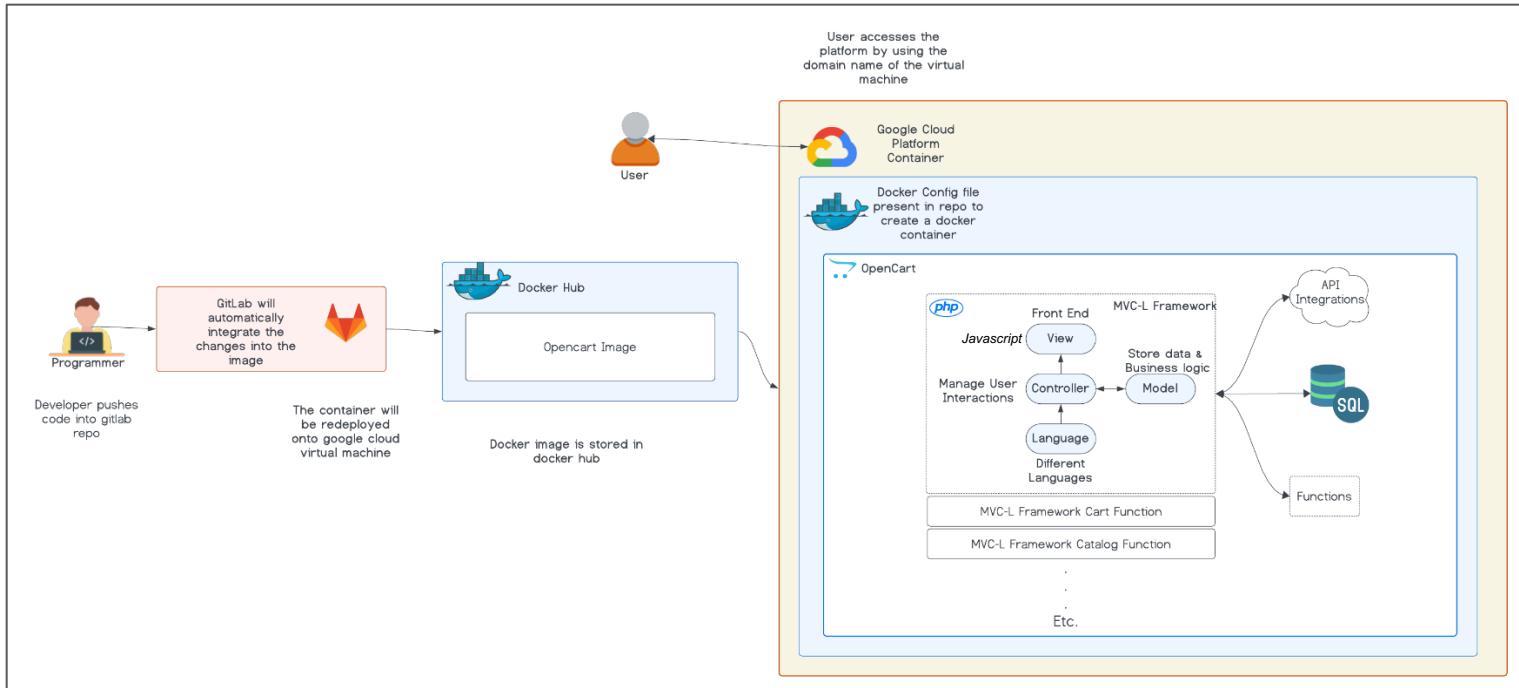


Figure 6 Overall Software Architecture

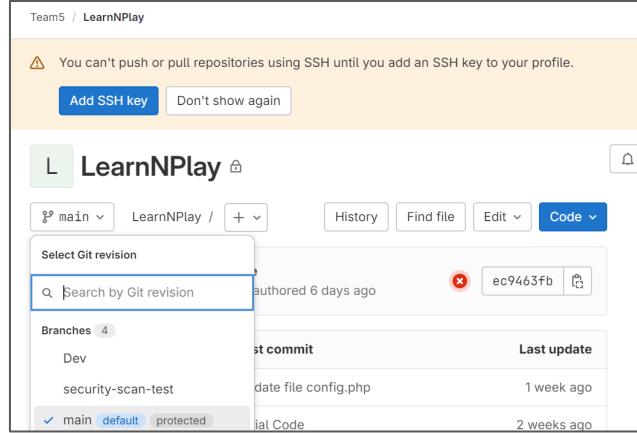
OpenCart Architecture

Our chosen project, OpenCart, uses a monolith architecture. Although this means that there are limitations to scaling this application, we believe that it is sufficient for our initial needs for the project, since we are not expecting a high load in the initial phase, and the storage required is not high as we have limited number of products in our catalogue. As the usage picks up, we have the following options to scale the application, depending on which is the bottleneck:

- scale vertically by increasing the size of the VM. This could alleviate any resource constraints temporarily while we scale out the main application infrastructure as below.
- Scale horizontally by spinning up multiple instances of the application and setting up a load balancer in front in case there is higher-than-expected web traffic.
 - In the event that there is unequal usage of the functions, we could consider breaking the code up into microservices to avoid over-reservation of resources that is unutilized, which causes a waste of money.
- Partition MySQL tables by date and add additional caching layer between the model and the database, to improve the query response time when there are many transactions in the database
- We chose to move our deployment from physical server to container because it would allow infrastructure-as-code and increase immutability of the server infrastructure. At the same time, it would allow more control over changes made to the deployment (through git merge and pull request approvals), and we would not have to rely on a super-admin to deploy the updated codes to the server. This also makes it possible for zero-downtime deployments in the future when we make code changes to implement the missing functionalities. We could also further improve the availability by deploying directly into Google's Kubernetes Engine so that we could utilize the auto-scaling and do blue-green deployment or rolling updates to achieve zero-downtime updates, something we did not yet manage to do due to time constraints.

Continuous Integration and Continuous deployment

We used *GitLab* for version control as well as to implement our CI/CD pipelines, so that we could collaborate remotely



and deliver smaller and frequent updates. As shown in

Figure 7 Branching we had initially created a development branch for individuals to merge their changes into. It was envisioned that this could allow us to deploy the changes into a common “development” instance, so that we could test out the deployed artifacts before the code was deployed into production. But due to limitations on the free credits and storage limitations on GCP, we decided to merge directly into the main branch , which would trigger the pipeline to deploy the codes for the production environment with an approval process, as shown in **Error! Reference source not found.**, which depicts one of our merges to main branch which requires an approval by a reviewer before the merge can be completed.

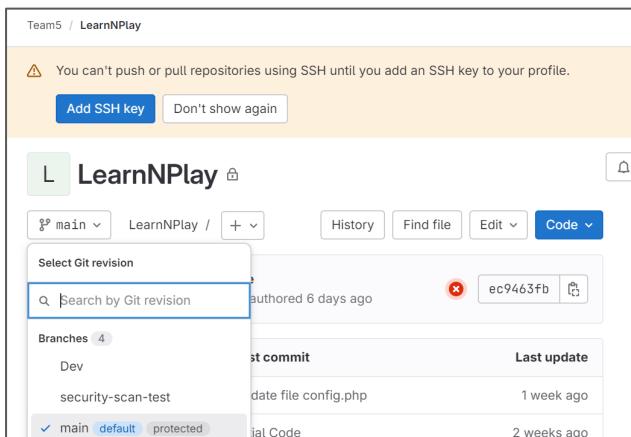


Figure 7 Branching

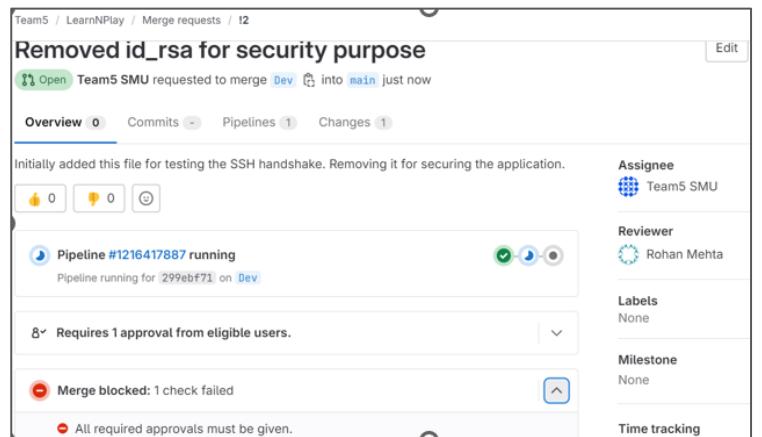


Figure 8 Merging with Approval

We have also implemented the pipelines using *gitlab-ci.yml* file. For CI, we specified 2 stages which are build and check. **Build** creates a docker container image of the application with all its dependencies like php, mysql ,etc. The dependencies are also installed using their public docker images. Then the **check** stage invokes the automated static code analysis that we had incorporated. Details will be covered in the Security Considerations section below. **Deploy** stage deploys this container image to the server and runs the image where the website shows all the latest updates. For running the *gitlab jobs*, we had to setup *gitlab-runner* on our local machine as we used the original dockerfile which could not be run on the publicly available gitlab runners. A screenshot of our *gitlab-ci.yml* file is shown in Figures 9 an 10 below which shows stages of our CI/CD process.

```

.gitlab-ci.yml
1 include:
2   - template: Jobs/SAST.gitlab-ci.yml
3
4 stages:
5   - build
6   - check
7   - deploy
8
9 variables:
10 SAST_EXCLUDED_ANALYZERS: "nodejs-scan,semgrep,sobelow,brakeman"
11
12 build:
13   stage: build
14   image: docker:latest
15 > services:...
16   script:
17     - docker build --no-cache -t priyah2021/learnplay:1.0.0 .
18     - docker push priyah2021/learnplay:1.0.0
19
20
21 phstan:
22   stage: check
23   before_script: []
24   allow_failure: true
25 > image: ...
26 > script: ...
27 > artifacts: ...
28
29 > sast:...
30
31 > deploy:
32   stage: deploy
33   script: ...
34

```

Figure 9 Docker file



Figure 10 CI-CD Pipeline

We have also used some other agile and devops practices, for example we are using infrastructure as a service for provisioning of the server to host this application. We are also using virtualization in the form of GCP VM instance, which we are connecting using the SSH handshake with public and private keys incorporating security into the process so that the VM is not accessed by unauthorized entities. We have also made use of containerization using docker images for deployment. A better solution for deployment of this would have been to use container as a service, but due to lack of familiarity and limited time for the purpose of this project, we stuck to technologies that the team was already familiar with.

```

PS C:\Users\priya> ssh priya_hlalla@34.87.20.190
Linux instance-20240220-152331.c.opencart-414915.internal 6.1.0-18-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.76-1 (2024-02-01) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Mar 12 08:52:46 2024 from 202.166.65.34
priya_hlalla@instance-20240220-152331:~$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
NAMES
64852fed8c01      priyah2021/learnplay:1.0.0    "docker-php-entrypoi..."   2 days ago        Up 2 days          0.0.0.0:80->80/tcp, :::80->80/tcp
                                         priya_hlalla-opencart-1
a9815bff68e3      redis:latest           "docker-entrypoint.s..."   2 days ago        Up 2 days          6379/tcp
935771d0b17f      memcached:latest       "docker-entrypoint.s..."   2 days ago        Up 2 days          11211/tcp
81650a3aab38      postgres:latest        "docker-entrypoint.s..."   2 days ago        Up 2 days          5432/tcp
807daf95a32d      mysql:5.7             "docker-entrypoint.s..."   2 days ago        Up 2 days          0.0.0.0:3306->3306/tcp, :::3306->3306/t
cp_33060/tcp      priya_hlalla-mysql-1
priya_hlalla@instance-20240220-152331:~$ |

```

docker images running on our GCP VM.

Automated tests

We were also able to write some unit tests using PHPUnit to check the catalogue item, as well as standalone end-to-end test using Cypress, to test the update of one of the product details. Both had the potential to be integrated into the CI/CD pipeline in future sprints.

Security Considerations

We reviewed the key security aspects of interest shown below and added the respective mitigations. In general, we added a few static code analysis tools to uncover several potential vulnerabilities that needed to be worked on in subsequent sprints.

Security Aspects	Data of Concern	Possible Vulnerabilities	Security Findings and current mitigations	Additional Mitigations
Confidentiality	<ul style="list-style-type: none"> User Details (password, email, address, payment details, preferences) 	<ul style="list-style-type: none"> SQL injection Insider attack (Database dumps) 	<ul style="list-style-type: none"> Passwords are currently hashed in OpenCart 	<ul style="list-style-type: none"> Implement audit trails and encryption of the database dumps
Integrity	<ul style="list-style-type: none"> Product listing and checkout price Website Defacement 	<ul style="list-style-type: none"> Javascript manipulations XSS 	<ul style="list-style-type: none"> Using semgrep-sast, we found potential XSS vulnerabilities in the datepicker widget. This would be reviewed in subsequent sprints. 	<ul style="list-style-type: none"> Continue to run Static Application Security Testing (SAST) and CVE scans routinely.
Availability	<ul style="list-style-type: none"> Order and payment details (for fulfilment) 	<ul style="list-style-type: none"> Database down due to server out of space or other reasons Server down or unable to scale to incoming requests, DOS 	<ul style="list-style-type: none"> The application was moved from physical server to run on docker in a VM in GCP, and this would increase resiliency and availability of the application. 	<ul style="list-style-type: none"> Have a Disaster Recovery Plan (see details below)

Figure 11 Security Considerations Reviewed

Details of our Business Continuity Plan is as follows:

- Given the low user traffic, we could set aside a daily downtime (from 2am – 4am) for the full daily backup of the database, with 4-hourly incremental backups that could be restored if needed
- To obtain RPO of 1 minute, all order and payments made would send an email to the customer as well as our admin email, to ensure trackability of these transactions even if database data was corrupted or lost.
- Logging of database transactions would also help us to sieve through if any important transactions occurred after the previous backup.

Static Application Security Testing

In the check stage of CICD, we have added three static code scanning tests: PHPStan, phpcs-security-audit, and semgrep-sast

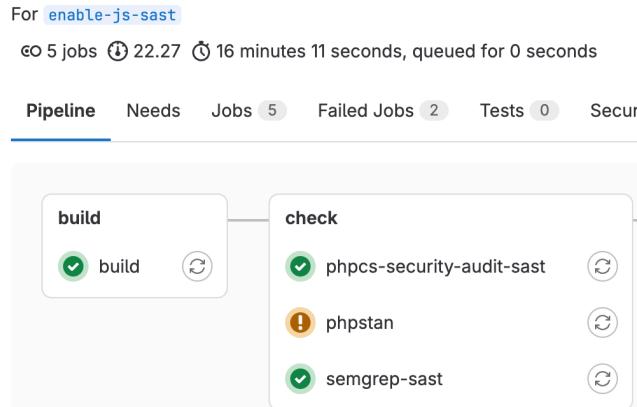


Figure 12 Screenshot showing check stage with phpstan, phpcs-security-audit-sast and semgrep-sast

We utilized the phpcs-security-audit and semgrep-sast analyser modules provided by GitLab to scan PHP and Javascript codes for vulnerabilities and security weaknesses. The figure below illustrates the results of one such scan conducted by the SAST analyzer in CICD. This white box testing approach helped us to identify the security issues early in the development stage in ci pipeline for the opencart application. It was noted that semgrep-sast also identified a potential XSS vulnerability in the javascript code for the datepicker widget. This vulnerability report warrants further testing in subsequent sprints to identify its real implications. Actual report details are in attached in gl-sast-reoprt(semgrep).json file.

Severity	Vulnerability	Identifier	Tool
High	Filesystem function dirname() detected with dynamic parameter directly from user input upload/install/index.php	PHPStan_SecurityAudit.Ba...	SAST GitLab
High	Filesystem function dirname() detected with dynamic parameter directly from user input system/config/default.php	PHPStan_SecurityAudit.Ba...	SAST GitLab
High	Please do not use eval() functions upload/system/storage/vendor/twig/twig/src/Environment.php	PHPStan_SecurityAudit.Ba...	SAST GitLab
High	Filesystem function dirname() detected with dynamic parameter directly from user input upload/install/index.php	PHPStan_SecurityAudit.Ba...	SAST GitLab

PHPStan was also used to enhance code quality by detecting errors and inconsistencies early in the development process. This proactive approach aids in maintaining a robust and error-free codebase. The figure below illustrates the results of one such scan conducted by PHPStan in CICD.

- Found 173 code quality issues
 - This report contains all Code Quality issues in the source branch.
- ▼ Major - Undefined variable: \$order
 - in upload/admin/controller/design/seo_url.php:611
- ▼ Major - Undefined variable: \$page
 - in upload/admin/controller/design/seo_url.php:612
- ▼ Major - Undefined variable: \$seo_url_id
 - in upload/admin/controller/design/seo_url.php:622
- ▼ Major - Undefined variable: \$seo_url_id
 - in upload/admin/controller/design/seo_url.php:632
- ▼ Major - Undefined variable: \$limit
 - in upload/admin/controller/design/seo_url.php:638
- ▼ Major - Undefined variable: \$page
 - in upload/admin/controller/design/seo_url.php:638
- ▼ Major - Undefined variable: \$limit
 - in upload/admin/controller/design/seo_url.php:639
- ▼ Major - Undefined variable: \$total
 - in upload/admin/controller/design/seo_url.php:641
- ▼ Major - Undefined variable: \$page
 - in upload/admin/controller/design/seo_url.php:644

CVE Scanning

Based on CVE Scanning, the OpenCart, version 4.0.2.3 which is used by the team for sprint 1, has a security vulnerability which allows authenticated backend users having common/security write privilege to write arbitrary untrusted data inside config.php and admin/config.php, resulting in remote code execution on the underlying server. However, the team also noted that a [hotfix](#) was available, which could be applied in subsequent sprints. Furthermore, the user had to have root access to exploit the CVE, so that CVE did not increase the risk likelihood nor impact. But it did highlight the need to implement proper audit logging to improve the auditability of the system..

Lessons Learnt

When we are evaluating potential software for adaptation, it is crucial that the evaluation be done in a more comprehensive manner, including not just the feature suitability, but also the CI/CD aspects (such as whether automated tests are available), and also security aspects (whether any security scanning tools are already in place, what are the scan results and rate of bug fixes etc), to ensure that the eventual product chosen is maintainable and extendable.

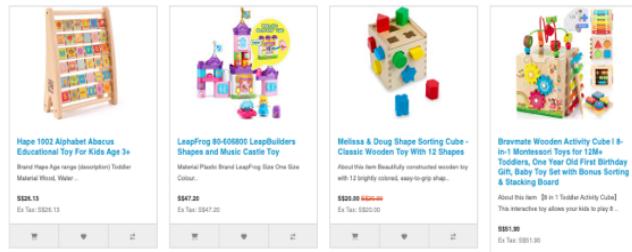
User Interface

The below figures show the User Interface of our website, with product details populated. We have customized the OpenCart application as per the requirement of our vision. Also here is the link for the website:
<http://shop.agiledevsecops.lol/>

SE Currency +65-12345678 My Account Wish List (0) Shopping Cart Checkout

LeapBuilders
Smart Blocks.
Smart Kids!

2+ years

Featured


SE Currency +65-12345678 My Account Wish List (0) Shopping Cart Checkout

Hape 1002 Alphabet Abacus Educational Toy For Kids Age 3+

\$26.13
Ex Tax: \$26.13

Add to Cart

Descriptions **Reviews (0)**

Brand: Hape
Age range (descriptive): Toddler
Material: Wood, Water based paint, PET
Colour: WOOD
Educational objective: Spacial Ability, Creativity, Imagination Development, Motor Skills
Size: 12" x 19" x 26"
Item weight: 3.00 kg
Item dimensions LxWxH: 30 x 29 x 12 centimeters
Operation mode: Manual

About this item

- Innovative design gives the ABC's a playful feel.
- Develops motor skills and stimulates imagination.
- It's safe.
- Print features are all non-toxic, child-safe, and of the highest quality.
- Dimensions include Baltic Birch, Maple and Piniferous wood from FSC-managed forests.

Technical Details

Educational Objective(s)	Spacial Ability, Creativity, Imagination Development, Motor Skills
Language	English, French
Model Number	Hape 1002
Assembly Required	No
Barcode	885625000000
Item Type	No
Material Type(s)	Wood, Water based paint, PET
Power Source Required	No
Color	WOOD
Dimensions	12" x 19" x 26"
Manufacturer recommended age	12 months - 5 years
Item model number	30 x 29 x 12 cm, 307.14 Grams
Product dimensions	30x29x12cm

Product description

Alphabet Abacus Toy is a�� distinctive abacus and its innovative design, gives the ABC's a playful feel with letters and pictures. Develops motor skills and stimulates imagination. Quality craftsmanship from Baltic Birch plywood and Piniferous. Children can easily sort and add and count the beads on the board. Letters are not paper laminated so they won't peel off.

Delight in Discovery with Alphabet Abacus by Hape

Brinware Wooden Activity Cube

Innovative design gives the ABC's a playful feel with the Hape Alphabet Abacus. The Abacus is a totally self-contained system of spinning wooden knobs that promote creativity, hand-eye coordination, and manipulations and introduces children to the concept of numbers. The abacus with numbered letters and numbers spinning on the Alphabet Abacus. Spin the face in corresponding order "A" on one side, "apple" on the other, "1" on one side and "one" and so on. Colors are very correct and designs are a little bit naive. Children love sorting the tiles and engaging with the letters. Helps with sorting and naming the animals and items depicted on each tile. Made from the durability of the maple wood blocks and natural wood particles. Hape uses safe screen printing right on the wood. Letters are not paper laminated so they won't peel off.

Brinware Wooden Activity Cube with Alphabet Abacus by Hape

Brinware Wooden Activity Cube

Innovative design gives the ABC's a playful feel with the Hape Alphabet Abacus. The Abacus is a totally self-contained system of spinning wooden knobs that promote creativity, hand-eye coordination, and manipulations and introduces children to the concept of numbers. The abacus with numbered letters and numbers spinning on the Alphabet Abacus. Spin the face in corresponding order "A" on one side, "apple" on the other, "1" on one side and "one" and so on. Colors are very correct and designs are a little bit naive. Children love sorting the tiles and engaging with the letters. Helps with sorting and naming the animals and items depicted on each tile. Made from the durability of the maple wood blocks and natural wood particles. Hape uses safe screen printing right on the wood. Letters are not paper laminated so they won't peel off.

Information Customer Service Extras My Account

Terms & Conditions Contact Us Brands Gift Certificates My Account

Delivery Information Returns Affiliate Newsletter

About Us Site Map

Privacy Policy

Powered By OpenCart Learn N Play © 2024

SE Currency +65-12345678 My Account Wish List (0) Shopping Cart Checkout

LeapBuilders
Smart Blocks.
Smart Kids!

2+ years

Appendix

Figures

The screenshot shows a Jira Backlog board for 'Team 5' under the 'Backlog' tab. The board is organized into two main sections: 'Sprint 2' (5 issues) and 'Backlog' (41 issues). Each issue is represented by a card with a summary, priority, progress, and assignee. The backlog items include user stories like 'T5-24 As a user who shops occasionally, I want to be able to check out as a guest without creating an account' and 'T5-17 As a user who wants to save time, I want to use my email address to register and login.' The board also includes sections for 'SHOPPING CART AND...', 'TECH OPERATIONS', and 'USER AUTHENTICATION'. A sidebar on the left provides navigation links for planning, development, and reporting.

Figure 13 Backlog showing more user story examples

The screenshot shows a detailed view of a user story titled 'As a sales rep, I want to adjust discount prices using my admin account to show users the pre- and post- discount prices so that they will be more happy to make the purchase.' The story has a priority of 'High' and is assigned to 'LI SHUANGSHUANG'. It is linked to parent story 'T5-32 Seller - Marketing and Promo' and is part of sprint 'None'. The 'Details' panel on the right lists the following fields: Priority (High), Assignee (LI SHUANGSHUANG), Labels (None), Parent (T5-32 Seller - Marketing and Promo), Sprint (None), Story point estimate (2), Fix versions (MVP), and Reporter (CHOW HUI LING _). The main content area contains the 'Description' and 'Acceptance Criteria' sections. The 'Description' section states: 'DOD: Test that promotion prices are displayed correctly in the product listing, document additional bugs/features as separate user story'. The 'Acceptance Criteria' section details the steps for testing the discount adjustment feature.

Figure 14 Example of DOD and Acceptance Criteria details captured in a user story

Wednesday < Feb 7, 2024 >

	Name	Priorities 😊	Progress 😊	Problems 😞
1	@Priya Lalla	NA (busy with other work)	Discussed the action items to be worked on for CI/CD	None
2	@LI SHUANGSHUANG	Define the categories of products	Define the user stories from seller and buyer perspectives	None
3	@CHOW HUI LING _	NA (busy with other work)	finished inputting user stories into Jira	None
4	@kruti chandrashekhar	Research on website names	came up with a unique title "Learn N Play"	None.
5	@twfoo.2022	Set up development server	Managed to install and run opencart on XAMPP	Various docker/nodejs/evershop issues

Figure 15 Confluence Daily Task Updates

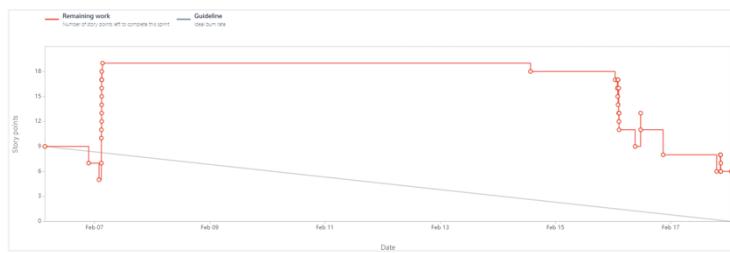


Figure 16 Sprint 1 Burndown Chart

Incomplete issues					
Key :	Summary :	Issue type :	Epic :	Status :	Assignee :
T5-17	As a user who wants to save time, I want to use my email address to register and login.	Story	USER AUTHENTICATION AN...	IN PROGRESS	
T5-24	As a user who shops occasionally, I want to be able to check out as a guest without c...	Story	SHOPPING CART AND CHEC...	WONT DO	
T5-82	Research on CI/CD Implementation from github	Story	TECH OPERATIONS	IN PROGRESS	

Completed issues					
Key :	Summary :	Issue type :	Epic :	Status :	Assignee :
T5-15	Finalise project vision	Story	TECH OPERATIONS	DONE	
T5-18	As a user who shops occasionally, I want to be able to browse products as a guest wit...	Story	SEARCH AND NAVIGATION	DONE	
T5-22	As a guest user, I can register for a user profile from the landing page	Story	USER AUTHENTICATION AN...	DONE	
T5-40	As a sales rep, I want to edit the product price using my admin account whenever the...	Story	SELLER - CATALOGUE MAN...	DONE	
T5-41	As a sales rep, I want to adjust discount prices using my admin account to show user...	Story	SELLER - MARKETING AND...	DONE	
T5-38	As a sales rep, I want to edit the product categories using my admin account so that ...	Story	SELLER - CATALOGUE MAN...	DONE	
T5-39	As a sales rep, I want to add the product details like descriptions and pictures using ...	Story	SELLER - CATALOGUE MAN...	DONE	
T5-45	As a user looking for convenience, I want to filter products by categories.	Story	SEARCH AND NAVIGATION	DONE	
T5-53	As a busy working parent, I want a quick and efficient checkout process to calculate ...	Story	SHOPPING CART AND CHEC...	DONE	
T5-65	Create the Software architecture diagram so that we know what are the components i...	Task	TECH OPERATIONS	DONE	
T5-66	Website Logo	Task	TECH OPERATIONS	DONE	
T5-67	Share latest code on with the team on MS Teams	Task	TECH OPERATIONS	DONE	
T5-81	Create a database diagram to understand data flow	Story	SUPPORT	DONE	

Figure 17 Completed and Incomplete Issues from Sprint 1

1.

Start doing	Stop doing	Keep doing
<ul style="list-style-type: none"> Measure team's capacity for the sprint based on the leaves/additional tasks, and accordingly plan the user stories for the sprint Split the user stories in more granular level so that the scope of work is better defined based on the sprint velocity Do daily standup updates via WhatsApp (While waiting for CI/CD to be setup fully) try to commit to git before making backend changes to the deployments, so that it is easier to revert to previous versions. 	<ul style="list-style-type: none"> Combining more than 1 requirement into one user story Daily standup updates on confluence due to lack of notifications, problems cannot be resolved immediately 	<ul style="list-style-type: none"> Share info about own tasks as a learning for the team Sharing updates on Whatsapp Helping each other in completing tasks Need a little longer sharing sessions for understanding how project is getting developed. for sprint review

✓ Action items

Add 1-2 follow-up action items to help the team apply what they learned in the retrospective:

- @Priya Lalla , @twfoo.2022 to explore github/gitlab options for implementing CI/CD on physical machine vs. on cloud (VM)
- @CHOW HUI LING _ to upload the mid term project into MS Teams

✓ Other items discussed

Add 1-2 follow-up action items to help the team apply what they learned in the retrospective:

- To shift themes to a later part of the project since it is a good to have and we didn't get it working the first time
- Should we use automated tests? → Not yet, will probably get more clarity after CICD is set up.
- Need to backup the database and prepare for recovery also before major code or configuration changes are made

Figure 18 Screenshot of Sprint Retrospective Minutes for Sprint 1

The screenshot shows a Jira Software interface for a bug report titled "When the user or admin tries to login, there is an error".

Project Navigation: Team 5 Software project, under PLANNING, Issues tab.

Bug Report Details:

- Summary:** When the user or admin tries to login, there is an error
- Assignee:** twfoo.2022 (with "Assign to me" button)
- Labels:** None
- Sprint:** Sprint 2
- Story point estimate:** 3
- Fix versions:** None
- Development:** None
- Releases:** None
- Reporter:** CHOW HUI LING _
- Checklist:** Open Checklist (checkbox checked)
- Activity:** Show: All, Comments, History, Estimate (async mode), Checklist history. Newest first.
- Comments:** Add a comment... (with a note: Pro tip: press M to comment)

Figure 19 Re-direction Bug identified in Sprint 1 due to website themes

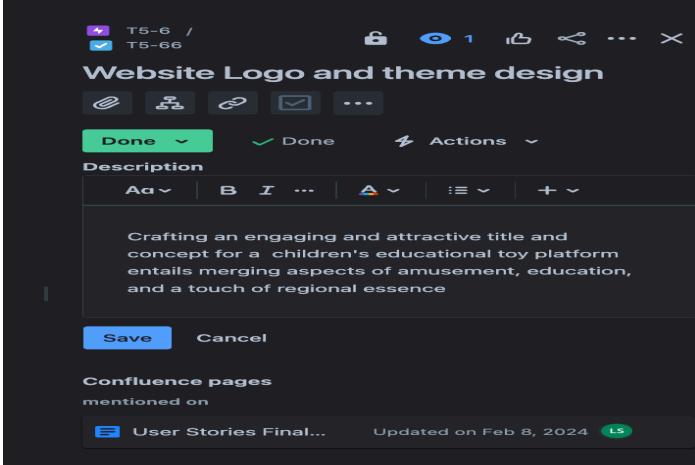


Figure 20 Splitting up of user story (Before)

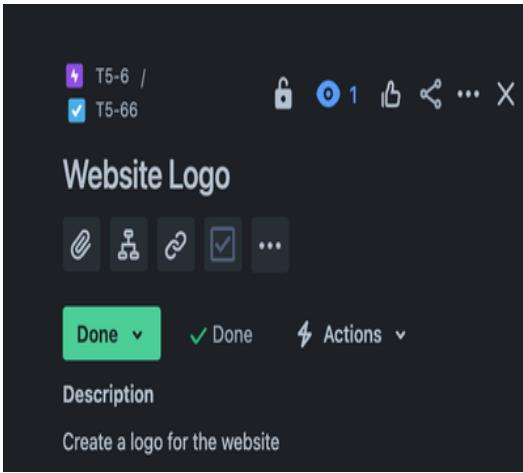


Figure 21 Splitting up of user story (after) marked as done in Sprint 1

Scope changes log						
Date	Key	Summary	Issue type	Epic	Details of scope change	Change in estimation
2024-02-17	T5-81	Create a database diagram to understand data flow	Story	SUPPORT	Estimate of 1 has been added	- → 1
2024-02-16	T5-82	Research on CI/CD implementation from github	Story	TECH OPERATI...	Estimate changed from 4 to 2	4 → 2
2024-02-16	T5-82	Research on CI/CD implementation from github	Story	TECH OPERATI...	Estimate of 4 has been added	- → 4
2024-02-16	T5-82	Research on CI/CD implementation from github	Story	TECH OPERATI...	Issue added to sprint	-
2024-02-16	T5-81	Create a database diagram to understand data flow	Story	SUPPORT	Issue added to sprint	-
2024-02-16	T5-45	As a user looking for convenience, I want to filter ...	Story	SEARCH AND N...	Estimate changed from 1 to 2	1 → 2
2024-02-16	T5-53	As a busy working parent, I want a quick and effic...	Story	SHOPPING CAR...	Estimate changed from 1 to 3	1 → 3
2024-02-16	T5-39	As a sales rep, I want to add the product details li...	Story	SELLER - CATA...	Estimate changed from 1 to 3	1 → 3
2024-02-16	T5-41	As a sales rep, I want to adjust discount prices usi...	Story	SELLER - MARK...	Estimate changed from 1 to 2	1 → 2
2024-02-16	T5-40	As a sales rep, I want to edit the product price usi...	Story	SELLER - CATA...	Estimate changed from 1 to 2	1 → 2
2024-02-16	T5-15	Finalise project vision	Story	TECH OPERATI...	Estimate changed from 2 to 1	2 → 1
2024-02-16	T5-15	Finalise project vision	Story	TECH OPERATI...	Estimate changed from 1 to 2	1 → 2
2024-02-16	T5-38	As a sales rep, I want to edit the product categori...	Story	SELLER - CATA...	Estimate changed from 1 to 2	1 → 2

Figure 22 Sprint report - Scope changes

The screenshot shows a Docker job log in a web-based interface. The log output is as follows:

```
1010 #12 0.180 To activate the new configuration, you need to run:  
1011 #12 0.180    service apache2 restart  
1012 #12 DONE 0.2s  
1013 #13 exporting to image  
1014 #13 exporting layers  
1015 #13 exporting layers 1.9s done  
1016 #13 writing image sha256:87766c3d79b2f1f7895eab113bb2a00ea8504eee6aafff186b6  
e5a483decc713 done  
1017 #13 naming to docker.io/priyah2021/learnplay:1.0.0 done  
1018 #13 DONE 1.9s  
1019 $ docker push priyah2021/learnplay:1.0.0  
1020 The push refers to repository [docker.io/priyah2021/learnplay]  
1021 78ae0f48483d: Preparing  
1022 73a6e61c5519: Preparing  
1023 1ecd11530bfe: Preparing
```

On the right side of the interface, there are several status indicators and links:

- Duration: 2 minutes 5 seconds
- Finished: 2 hours ago
- Queued: 0 seconds
- Timeout: 1h (from project) [?](#)
- Runner: #12270845 (JLgUopmMV) 1-green.saas-linux-small-amd64.runners-manager.gitlab.com/default
- Commit: 63298c8a [🔗](#)
Merge branch 'Dev' into 'main'
- Pipeline #1216480356 [✖ Failed](#) for main [🔗](#)

Figure 23 Pushing image to Docker